5-21-2012

Trust Networks

Krishnaprasad Thirunarayan
Wright State University - Main Campus, t.k.prasad@wright.edu

Pramod Anantharam
Wright State University - Main Campus

Cory Andrew Henson
Wright State University - Main Campus

Amit P. Sheth
Wright State University - Main Campus, amit@sc.edu

Trust Networks

Krishnaprasad Thirunarayan (T. K. Prasad)

Professor, Department of Computer Science and Engineering

Kno.e.sis - Ohio Center of Excellence in Knowledge-enabled Computing

Wright State University, Dayton, OH-45435

(Collaborators: Pramod Anantharam, Cory Henson, and Professor Amit Sheth)

CTS-2012

Broad Outline

- Real-life Motivational Examples (Why?)
- Trust : Characteristics and Related Concepts (What?)
- Trust Ontology (What?)
  - Type, Value, Process, Scope
- Gleaning Trustworthiness (How?) + Robustness to Attack
  - Practical Examples of Trust Metrics
  - Comparative Analysis of Bayesian Approaches to Trust
- Research Challenges (Why-What-How?)
  - Sensor Networks
  - Social Networks
  - Interpersonal
- Details of Bayesian Approach to Multi-level Trust

Real-life Motivational Examples

(Why track trust?)

Interpersonal

- With which neighbor should we leave our children over the weekend when we are required to be at the hospital?
- Who should be named as a guardian for our children in the Will?

Social

- In Email:
  - SUBJECT: [TitanPad] Amit Sheth invited you to an EtherPad document.
  - CONTENT: View it here:
    http://knoesis.titanpad.com/200

- *Issue*: Is the request genuine or a trap?

Social

- To click or not to click a http://bit.ly-URL

- To rely or not to rely on a product review (when only a few reviews are present, or the reviews are conflicting)?

Sensors

- Weather sensor network predicts a potential tornado in the vicinity of a city.
- *Issue*: Should we mobilize emergency response teams ahead of time?
- Van's TCS (Traction Control System) indicator light came on intermittently, while driving.
- *Issue*: Which is faulty: the indicator light or the traction control system?
- Van's Check Engine light came on, while driving.
- *Issue*: Which is faulty: the indicator light or the transmission?

Man-Machine Hybrid Collaborative Systems

The 2002 **Uberlingen Mid-air Collision** (between Bashkirian Airlines Flight 2937 and DHL Flight 611) occurred because the pilot of one of the planes **trusted** the human air traffic controller (who was *ill-informed about the unfolding situation*), instead of the electronic TCAS system (which was providing *conflicting but correct course of action* to avoid collision).

http://en.wikipedia.org/wiki/2002_Uberlingen_mid-air_collision

Man-Machine Hybrid Collaborative Systems

In hybrid situations, artificial agents should reason about the trustworthiness and deceptive actions of their human counter parts. People and agents in virtual communities will deceive, and will be deceived.

Castelfranchi and Tan, 2002

Common Issues and Context

- Uncertainty
  - About the validity of a claim or assumption
  - Past Experience : Vulnerability

- Need for action
- Critical decision with potential for loss

Commonality among Trust Definitions*

- a Trustor
  - someone who must choose whether, and how much, to trust
- a Trustee
  - someone or something that is to be trusted
- an Action
  - by which the trustor is choosing to be vulnerable to the trustee based on an assessment of trustee's nature
- a Context
  - in which the potential negative consequences of betrayal outweigh any perceived positive results.

*http://www.iarpa.gov/rfi_trust.html

Sources of uncertainty

- In social network, trustor's incomplete knowledge or trustee's devious intentions.
- In sensor network, unpredictable environment (e.g., random phenomenon), or sensor faults (e.g., due to aging), leading to corrupt data.
- In cognitive radio networks, unpredictable channel noise or traffic.
- Interpersonal examples
  - Irresponsible / selfish / greedy / vengeful traits

Why Track Trust?

- In Mobile Ad Hoc Networks (MANETs), trust enables dynamic determination of secure routes.
  - *Efficiency*: To improve throughput
    - By avoiding nodes facing bad channel condition
  - *Robustness* : To detect malicious nodes
    - When attackers enter the network in spite of secure key distribution/authentication

Why Track Trust?

- In sensor networks, it allows detection of faults and transient bad behaviors due to environmental effects.

- In cognitive radio networks, it can enable selection of optimal channel (less noisy, less crowded channels).

Why Track Trust?

- In E-commerce:
  - To predict future behavior in a reliable manner.
  - To incentivize “good” behavior and discourage “bad” behavior.
  - To detect malicious entities.

The Two Sides of Trust

- *Trustor* assesses *trustee* for dependability.
- *Trustee* casts itself in positive light to *trustor*.
- **Trust** is a function of *trustee's* perceived trustworthiness and the *trustor's* propensity to trust.

Risk/uncertainty mitigation

- Compensating factors *that alter* ***trust threshold***
  - In e-commerce, warranties and insurance reduce risk.
  - In sensor networks, redundancy enables filtering of corrupt data.
  - In interpersonal situations, close ties help.

Trust and Related Concepts

(What is trust?)

Trust Definition : Psychology slant

Trust is the psychological state comprising a willingness to be vulnerable in expectation of a valued result.

Ontology of Trust, Huang and Fox, 2006
Josang et al's Decision Trust

Trust Definition : Psychology slant

Trust in a person is a *commitment to an action* based on a *belief* that the future actions of that person will lead to good outcome.

Golbeck and Hendler, 2006

Trust Definition : Probability slant

Trust (or, symmetrically, distrust) is a level of subjective probability with which an agent assesses that another agent will perform a particular action, both before and independently of such an action being monitored …

Can we Trust Trust?, Diego Gambetta, 2000
Josang et al's Reliability Trust

Trustworthiness Definition : Psychology Slant

Trustworthiness is a collection of qualities of an agent that leads them to be considered as deserving of trust from others (in one or more environments, under different conditions, and to different degrees).

http://www.iarpa.gov/rfi_trust.html

Trustworthiness Definition : Probability slant

Trustworthiness is the objective probability that the trustee performs a particular action on which the interests of the trustor depend.

Solhaug et al, 2007

Trust vs Trustworthiness : My View

Trust Disposition

Depends on

Potentially Quantified Trustworthiness Qualities

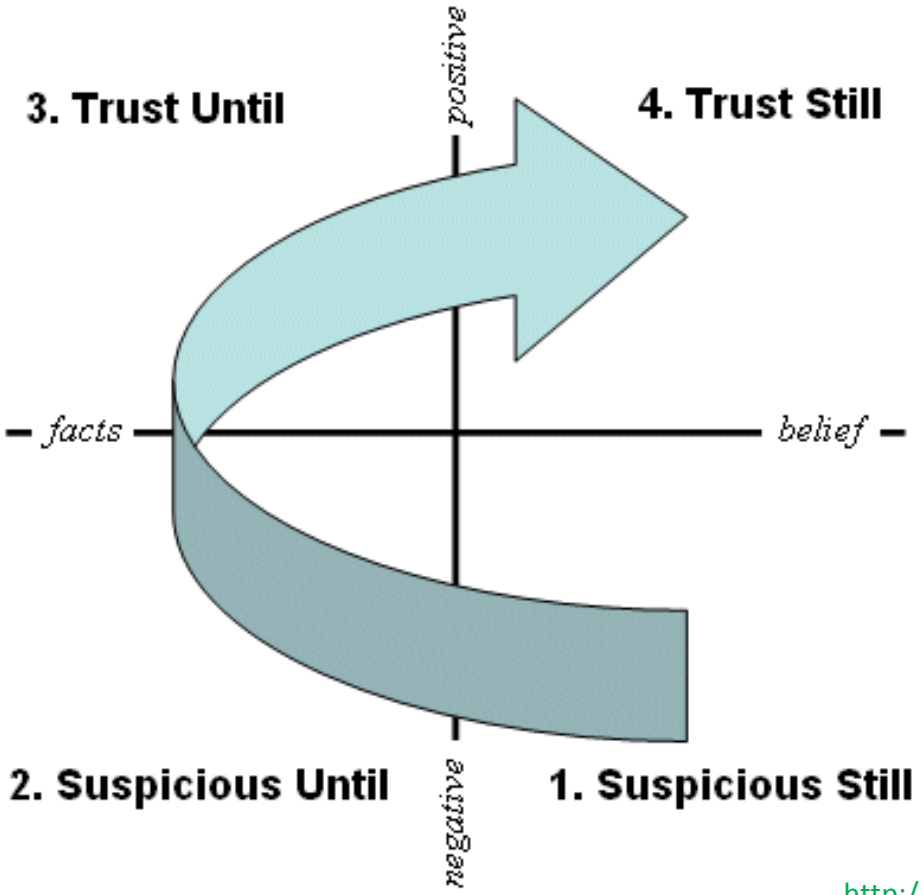
+

Context-based Trust Threshold

E.g.*, In the context of trusting strangers, people in the West will trust for lower levels of trustworthiness than people in the Gulf.

*Bohnet et al, 5/2010

Qualitative Trust Dispositions (depends on Trust Threshold)



http://paulenglish.com/trust.html

(Community-based) Reputation

- Reputation* is the community or public estimation of standing for merit, achievement, reliability, etc.

*dictionary.com

- Reputation** is the opinion (or a social evaluation) of a community toward a person, a group of people, or an organization on a certain criterion.

**Wikipedia

  - Cf. Brand-value, PageRank, eBay profile, etc.

Trust vs. (Community-based) Reputation

Reputation can be a basis for trust. However, they are different notions*.

- I trust you because of your good reputation.
- I trust you despite your bad reputation.
- Do you still trust Toyota brand?

*Josang et al, 2007

Trust vs. (Community-based) Reputation

Trust :: Reputation

::::

Local :: Global

::::

Subjective :: Objective

(Cf. *Security* refers to resistance to attacks.)

Reputation is Overloaded

Community-based Reputation

vs.

Temporal Reputation

(Cf. Sustained good behavior over time elicits *temporal reputation-based trust*.)

Trust is well-known,

but is not well-understood.

The utility of a notion testifies not to its clarity but rather to the philosophical importance of clarifying it.

-- *Nelson Goodman*

(Fact, Fiction and Forecast, 1955)

Trust Ontology

(What is trust?)

Illustration of Knowledge Representation and Reasoning:
Relating Semantics to Data Structures and Algorithms

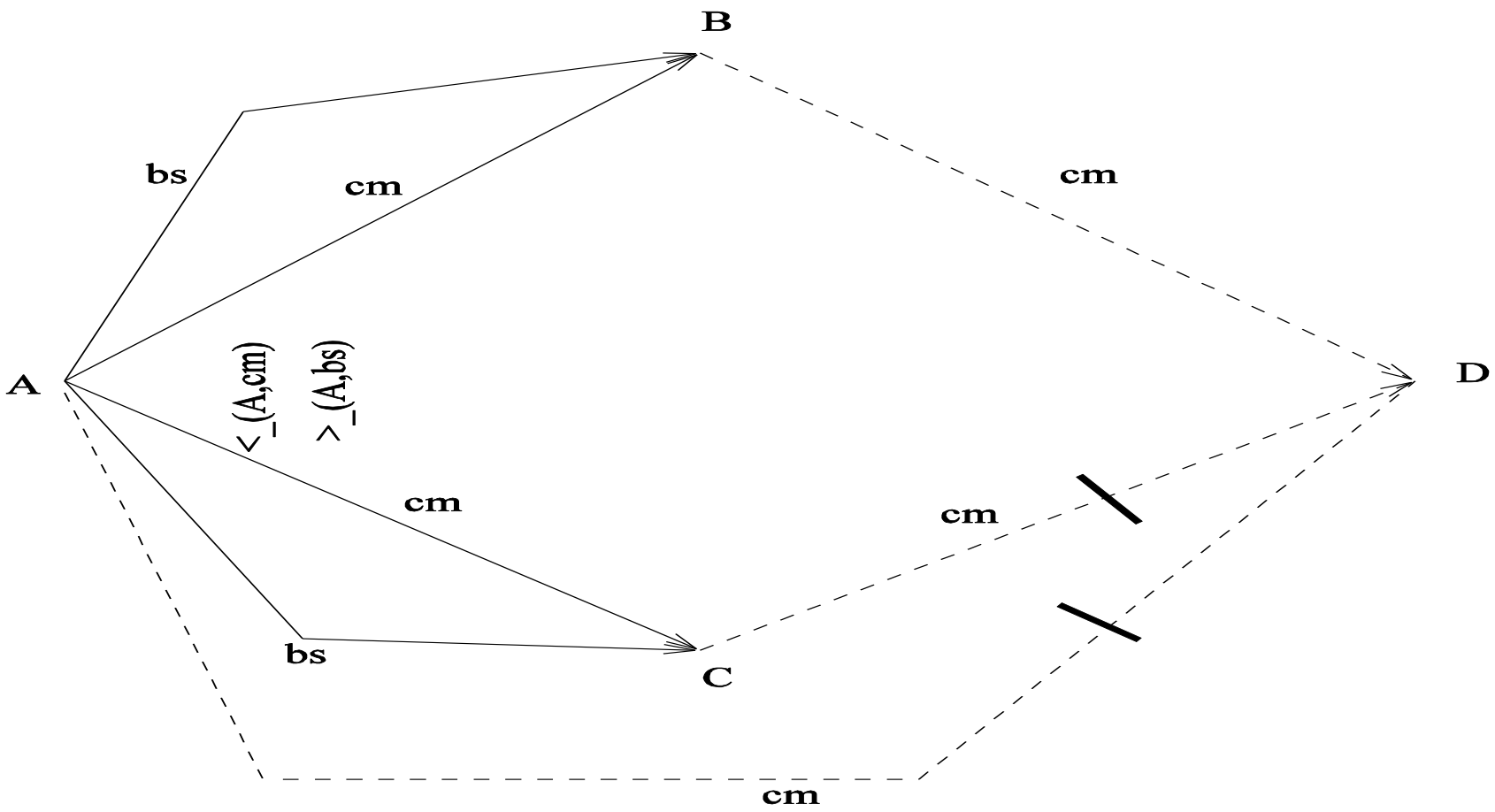
Example Trust Network - Different Trust Links with Local Order on out-links

- Alice trusts Bob *for recommending* good car mechanic.
- Bob trusts Dick *to be* a good car mechanic.
- Charlie *does not* trust Dick to be a good car mechanic.
- Alice trusts Bob *more than* Charlie, *for recommending* good car mechanic.
- Alice trusts Charlie *more than* Bob, *for recommending* good baby sitter.

*Thirunarayan et al, IICAI 2009

Digression: Illustration of Knowledge Representation and Reasoning

- Abstract and encode clearly delineated "subarea" of knowledge in a formal language.
  - Trust Networks => node-labeled, edge-labeled directed graph (DATA STRUCTURES)
- Specify the meaning in terms of how "network elements" relate to or compose with each other.
  - Semantics of Trust, Trust Metrics => using logic or probabilistic basis, constraints, etc. (SEMANTICS)
- Develop efficient graph-based procedures
  - Trust value determination/querying (INFERENCE ALGORITHMS)



Referral trust link (In recommendations)

Functional trust link (For capacity to act)

Nonfunctional trust link (For lack of capacity to act)

Trust Ontology*

6-tuple representing a trust relationship:

Type – Represents the nature of trust relationship.

Value – Quantifies trustworthiness for comparison.

Scope – Represents applicable context for trust.

Process – Represents the method by which the *value* is created and maintained.

*Anantharam et al, NAECON 2010

Trust Ontology:
Trust Type, Trust Value, and Trust Scope

- Trust Type*
  - *Referral Trust* – Agent a1 trusts agent a2's ability to recommend another agent.
  - (Non-)*Functional Trust* – Agent a1 (dis)trusts agent a2's ability to perform an action.
    - Cf. ** trust in belief vs. trust in performance
- Trust Value
  - E.g., Star rating, numeric rating, or partial ordering.
- Trust Scope*
  - E.g., Car Mechanic context.

*Thirunarayan et al, IICAI 2009
** Huang and Fox, 2006

Multidimensional / Orthogonal Trust Scopes in Ecommerce

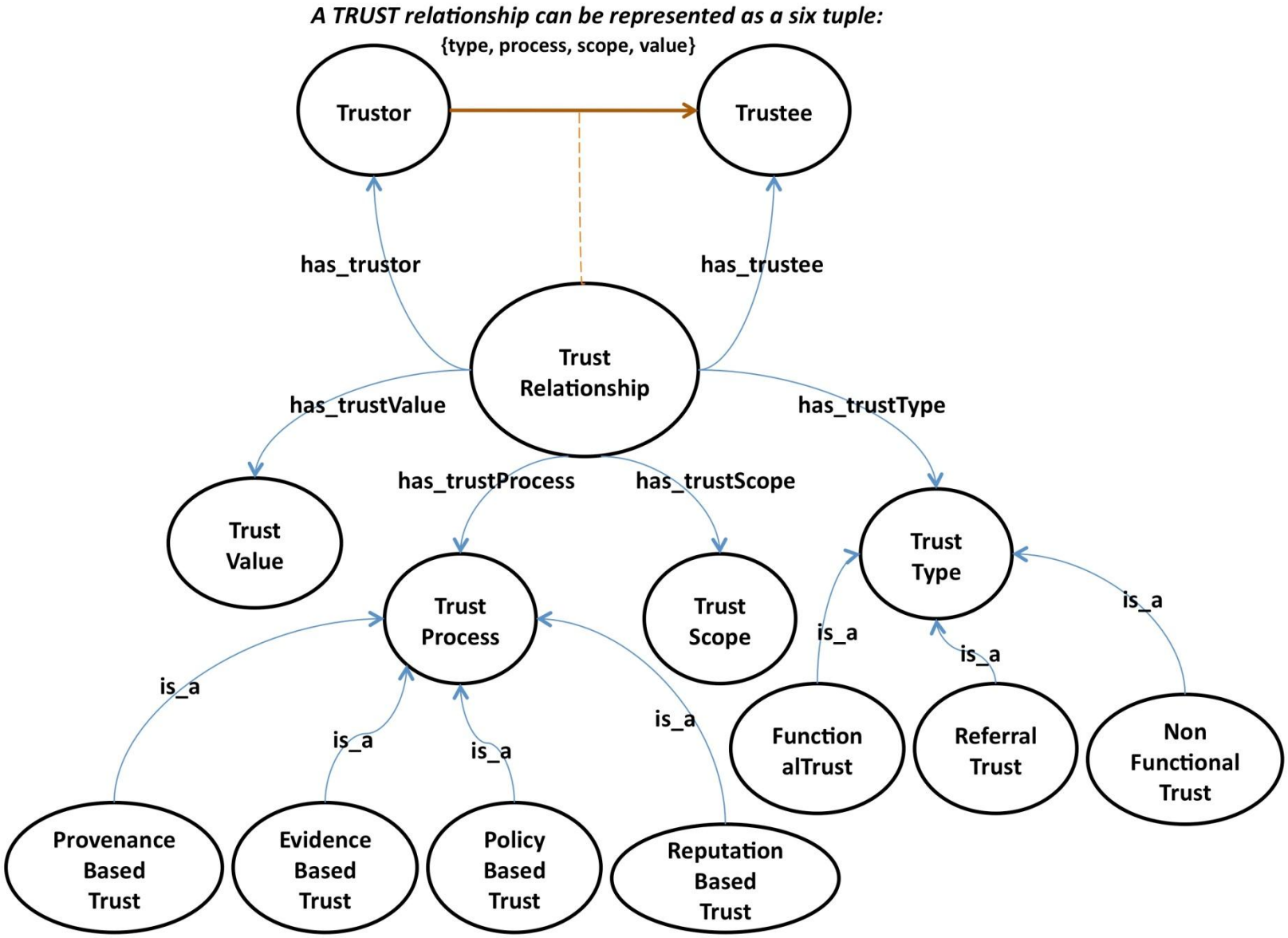
- Trust in a vendor to deliver on commitments
- Trust in vendor's ethical use of consumer data
- Trust in Internet communication being secure.
- Plus: Propensity/Disposition to trust

Trust Ontology:
Trust Process

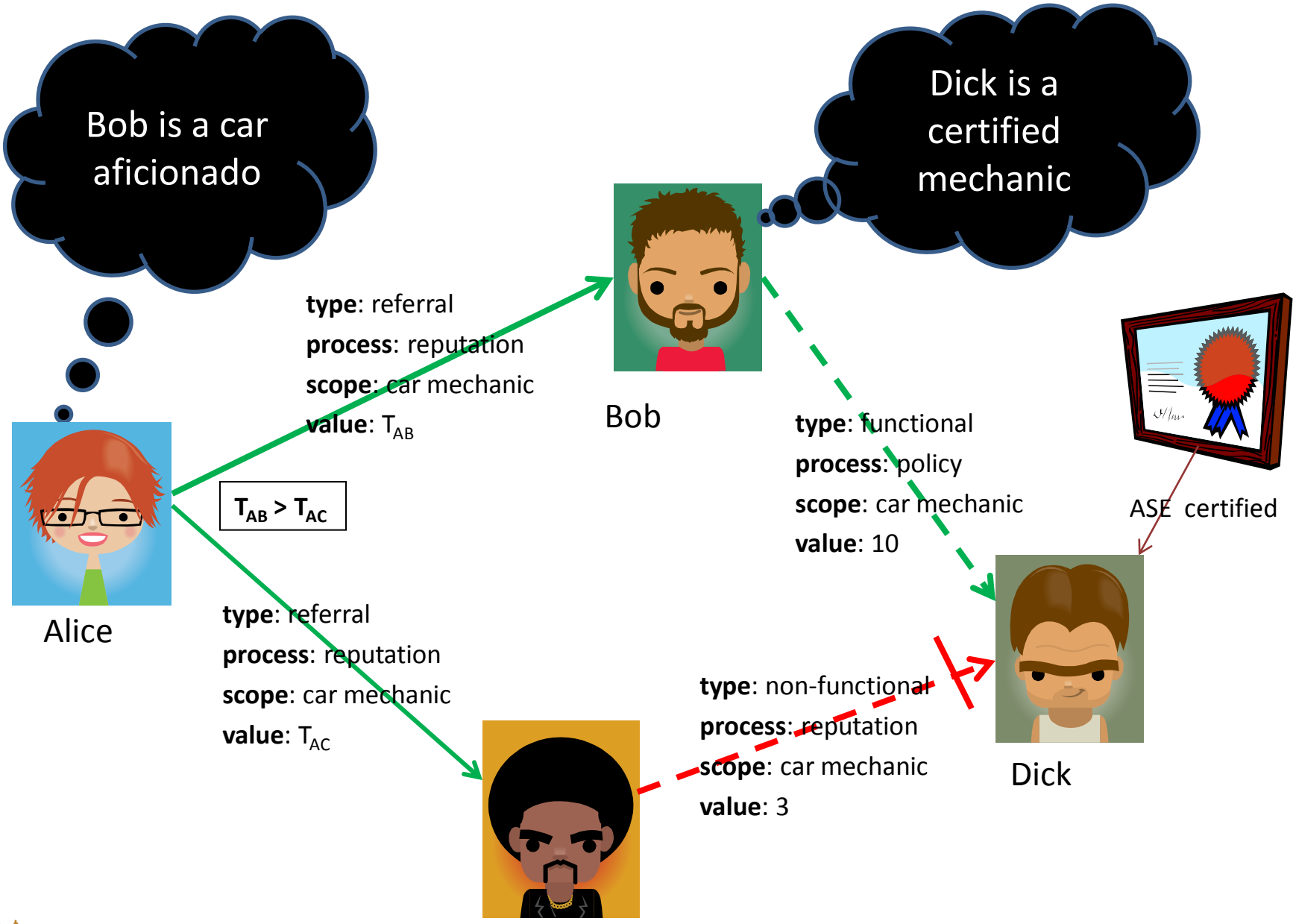
- Represents the method by which the value is computed and maintained.
  - **Primitive (for functional and referral links)***
    - Reputation – based on past behavior (temporal) or community opinion.
    - Policy – based on explicitly stated constraints.
    - Evidence – based on seeking/verifying evidence.
    - Provenance – based on lineage information.
  - **Composite (for admissible paths)****
    - Propagation (Chaining and Aggregation)

*Anantharam et al, NAECON 2010
**Thirunarayan et al, IICAI 2009

Trust Ontology



Example Trust Network illustrating Ontology Concepts



Unified Illustration of Trust Processes

Scenario : Hiring Web Search Engineer - An R&D Position

Various Trust Processes :

- (Temporal-based) Reputation: Past job experiences
- (Community-based) Reputation: Multiple references
- Policy-based: Score cutoffs on screening test
- Provenance-based: Department/University of graduation
- Evidence-based: Multiple interviews (phone, on-site, R&D team)

Deception

- Deception is the betrayal of trust.
  - Ironically, trust makes us prone to deception.
  - Knowing what features are used to glean trustworthiness can also assist in avoiding detection while deceiving.

Gleaning Trustworthiness : Practical Examples

(How to determine trustworthiness?)

Trust Metric and Trust Model

- Trust Metric => How is *primitive* trust represented?
  - E.g., Real number, Finite levels, Partial Order, Opinion = (belief, disbelief, uncertainty), etc.
- Trust Model => How is *composite* trust computed or propagated?

Y. L. Sun, et al, 2/2008

Ideal Approach

- Capture semantics of trust using
  - axioms for trust propagation, or
  - catalog of example trust networks that are equivalent.

- Develop trust computation rules for propagation (that is, chaining and aggregation) that satisfy the axioms or equivalence relation.

Direct Trust : Functional and Referral Reputation-based Process

(Using large number of observations)

Reputation-based Frameworks

- Centralized Trust Authority
  - E.g., E-commerce systems, etc.
- Distributed Trust Representation and Computation (using Bayesian analytics)
  - E.g., MANETs, peer-to-peer networks, etc.

Using Large Number of Observations

- Over time (<= Referral + Functional) :
  Temporal Reputation-based Process
  - Mobile Ad-Hoc Networks
  - Sensor Networks
    - Quantitative information
      (Numeric data)
- Over agents (<= Referral + Functional) :
  Community Reputation-based Process
  - Product Rating Systems
    - Quantitative + Qualitative information
      (Numeric + text data)

Desiderata for Trustworthiness Computation Function

- Initialization Problem : How do we get *initial* value?
- Update Problem : How do we reflect the *observed behavior* in the current value *dynamically*?
- Trusting Trust* Issue: How do we mirror *uncertainty* in our estimates as a function of observations?
  - Law of Large Numbers: The *average* of the results obtained from a large number of trials should be close to the *expected value*.
- Efficiency Problem : How do we *store* and *update* values *efficiently*?

*Ken Thompson's Turing Award Lecture: "**Reflections on Trusting Trust**"

Mathematical Background

Beta PDF for Reputation

Beta-distribution : Gently

- Consider a (potentially unfair) coin that comes up HEADS with probability p and TAILS with probability $(1 - p)$.
- Suppose we perform $(r + s)$ coin tosses and the coin turns up with HEADS r times and with TAILS s times.
- What is the best estimate of the distribution of the probability p given these observations?

  => Beta-distribution with parameters $(r+1, s+1)$

$$\mathbf{f}(p; r+1, s+1)$$

Beta Probability Density Function(PDF)

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,du}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\,x^{\alpha-1}(1-x)^{\beta-1}$$

$$= \frac{1}{\mathrm{B}(\alpha,\beta)}\,x^{\alpha-1}(1-x)^{\beta-1}$$

$$\mathrm{E}(X) = \frac{\alpha}{\alpha+\beta}$$

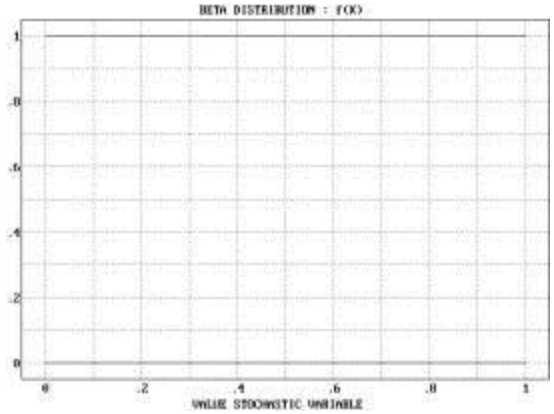
$$\mathrm{E}(X^2) = \frac{\alpha(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)}$$

$$\mathrm{Var}(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

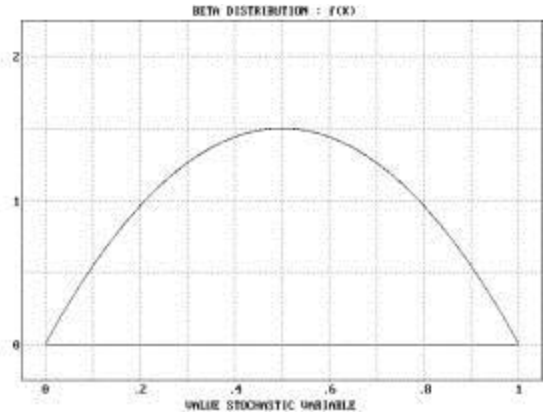
x is a probability, so it ranges from 0-1

If the prior distribution of x is uniform, then the beta distribution gives posterior distribution of x after observing $\alpha-1$ occurrences of event with probability x and $\beta-1$ occurrences of the complementary event with probability (1-x).

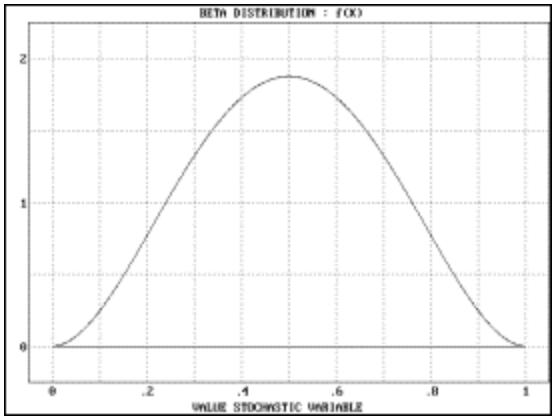
$\alpha = \beta$, so the pdf's are symmetric w.r.t 0.5.
Note that the graphs get narrower as $(\alpha+\beta)$ increases.



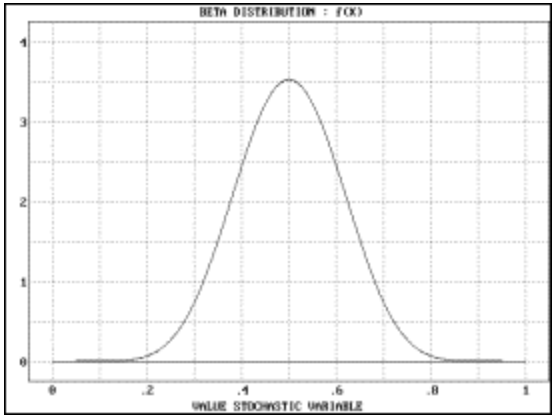
$\alpha = 1$
$\beta = 1$



$\alpha = 2$
$\beta = 2$

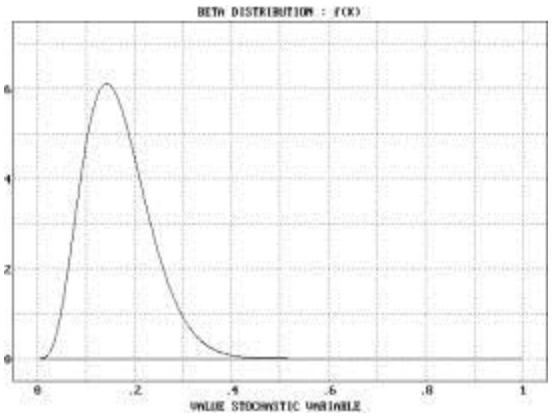


$\alpha = 5$
$\beta = 5$

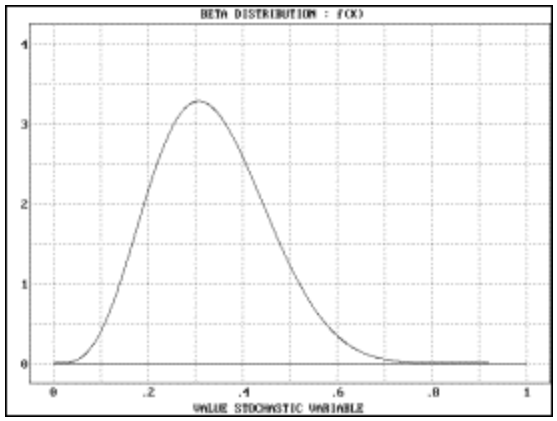


$\alpha = 10$
$\beta = 10$

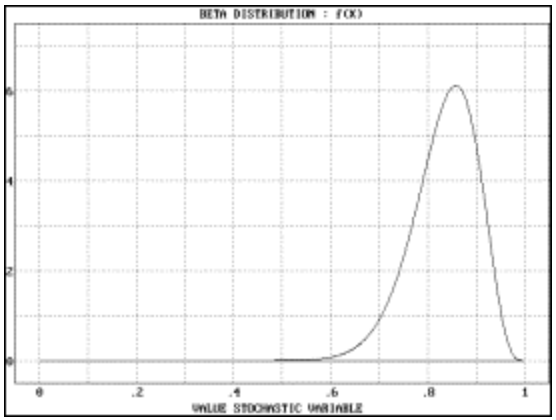
$\alpha \neq \beta$, so the pdf's are asymmetric w.r.t . 0.5.
Note that the graphs get narrower as $(\alpha+\beta)$ increases.



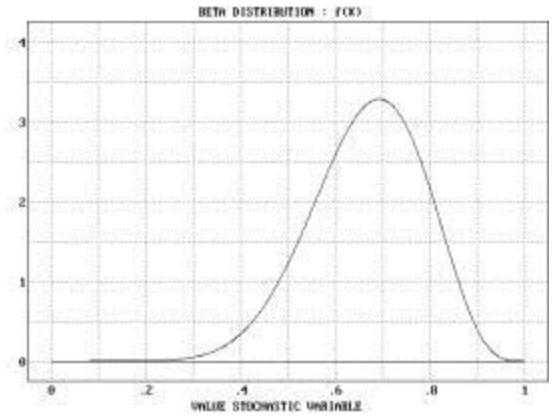
$\alpha = 5$
$\beta = 25$



$\alpha = 5$
$\beta = 10$



$\alpha = 25$
$\beta = 5$



$\alpha = 10$
$\beta = 5$

Beta-distribution - Applicability

- Dynamic trustworthiness can be characterized using beta probability distribution function gleaned from total number of correct (supportive) $r = (\alpha - 1)$ and total number of erroneous (opposing) $s = (\beta - 1)$ observations so far.
- Overall trustworthiness (reputation) is its mean: $\alpha/\alpha + \beta$

Why Beta-distribution?

- Intuitively satisfactory, Mathematically precise, and Computationally tractable
  - Initialization Problem : Assumes that all probability values are equally likely.
  - Update Problem : Updates (α, β) by incrementing α for every correct (supportive) observation and β for every erroneous (opposing) observation.
  - Trusting Trust Issue: The graph peaks around the mean, and the variance diminishes as the number of observations increase, if the agent is well-behaved.
  - Efficiency Problem: Only two numbers stored/updated.

Information Theoretic Interpretation of Trustworthiness Probability

- Intuitively, probability values of 0 and 1 imply certainty, while probability value of 0.5 implies a lot of uncertainty.

- This can be formalized by mapping probability in [0,1] to trust value in [–1,1], using information theoretic approach.

Y. L. Sun, et al, 2/2008

Information Theoretic Interpretation of Trustworthiness Probability

- T(trustee : trustor, action) =

  if 0.5 <= p

  then 1 – H(p) /* 0.5 <= p <= 1 */

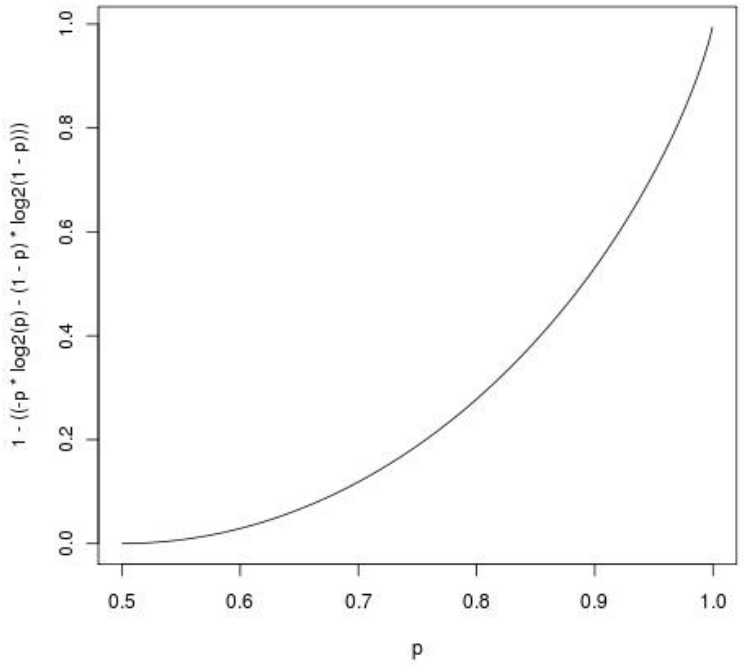
  else H(p) – 1 /* 0 <= p <= 0.5 */

  where

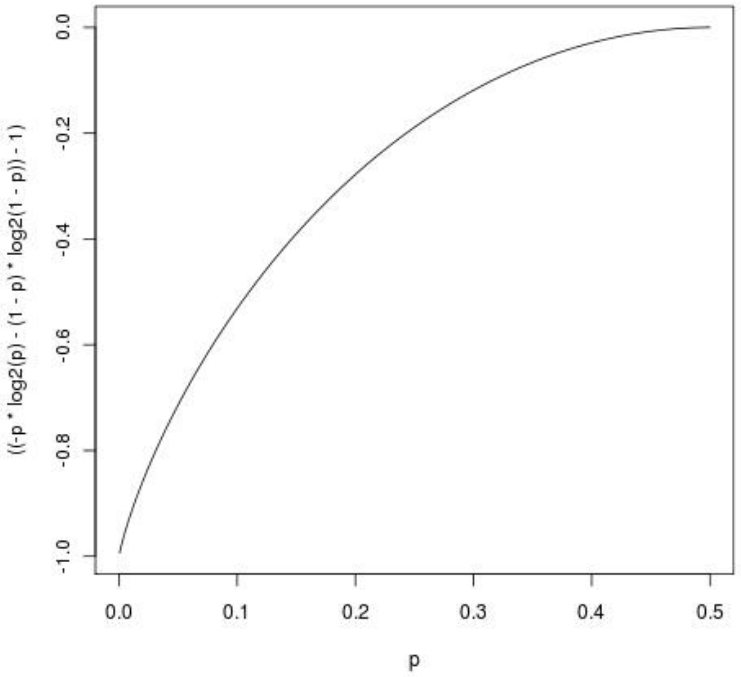
$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$$

Plot of T(trustee : trustor, action) vs. p

Trust portion (p in [0.5,1])



Distrust portion (p in [0,0.5])



Linear vs Nonlinear Map

Relative to computing trust as

$$T(\text{trustee} : \text{trustor}, \text{action}) = (p - 0.5) * 2$$

to map of trust probability p in [0,1] to a trust value in [-1,+1], the information theoretic formulation yields a non-linear map that amplifies the effect of changes to trust probability on the trust value at the extremes.

Direct Trust : Functional
Policy-based Process

(Using Trustworthiness Qualities)

General Approach to Trust Assessment

- Domain dependent qualities for determining trustworthiness
  - Based on Content / Data
  - Based on External Cues / Metadata
- Domain independent mapping to trust values or levels
  - Quantification through abstraction and classification

Example: Wikipedia Articles

- Quality (content-based)
  - Appraisal of information provenance
    - References to peer-reviewed publication
    - Proportion of paragraphs with citation
  - Article size
- Credibility (metadata-based)
  - Author connectivity
  - Edit pattern and development history
    - Revision count
    - Proportion of reverted edits - (i) normal (ii) due to vandalism
    - Mean time between edits
    - Mean edit length.

Sai Moturu, 8/2009

(cont'd)

- Quantification of Trustworthiness
  - Based on Dispersion Degree Score
    (Extent of deviation from mean)
- Evaluation Metric
  - Ranking based on trust level (determined from trustworthiness scores), and compared to gold standard classification using Normalized Discounted Cumulative Gain (NDCG)
    - RATINGS: featured, good, standard, cleanup, and stub.
    - NDCG: penalizes more heavily errors at the top.

Example: Websites

- Trustworthiness estimated based on criticality of data exchanged.
  - Email address / Username / password
  - Phone number / Home address
  - Date of birth
  - Social Security Number / Bank Account Number
- *Intuition*: A piece of data is critical if and only if it is exchanged with a small number of highly trusted sites.

Indirect Trust : Referral + Functional
Variety of Trust Metrics and Models

(Using Propagation – Chaining and Aggregation over Paths)

Collaborative Filtering

- Collaborative Filtering: Item-rating by a user predicted on the basis of user's similarity to other users.

- Similarity Measures:
  - Profile-based
  - Item-ratings-based
  - Item-category-based

Collaborative Filtering

- Pros:
  - Items-agnostic
  - Scales well over time with large number of items
- Cons:
  - Data Sparsity Problem: Small number of common items between users.
  - Cold Start Users: Small number of items rated by a user.
  - Prone to Copy-Profile Attack: An attacker can create a targeted-user-like profile to manipulate recommendations.

Massa-Avesani, 2007

Trust-aware Recommender System

- TaRS uses explicit/direct trust between users to predict implicit/indirect trust between users through chaining.
- Collaborative Filtering Limitations Overcome:
  - Mitigates Data Sparsity: Trust propagation is more general and improves coverage.
  - Bootstraps Cold Start Users: A single trust link from a new user can enable the user to inherit several "parental" recommendations.
  - Robust w.r.t Copy-Profile Attack: Fake identities are not trusted by an active user.

Massa-Avesani, 2007

Trust Propagation Frameworks

- **Chaining, Aggregation, and Overriding**

Golbeck – Hendler, 2006

Massa-Avesani, 2005
Bintzios et al, 2006

Sun et al, 2006
Thirunarayan et al, 2009

- Trust Management
  - Abstract properties of operators

Richardson et al, 2003

- Reasoning with trust
  - Matrix-based trust propagation

Guha et al., 2004

- The Beta-Reputation System
  - Algebra on opinion = (belief, disbelief, uncertainty)

Josang and Ismail, 2002

Trust Propagation Algorithms

- Top-down
  - 1: Extract trust DAG (eliminate cycles)
  - 2: Predict trust score for a source in a target by aggregating trust scores in target inherited from **source's** "trusted" parents weighted with trust value in the corresponding parent.
    - Computation is level-by-level
    - Alternatively, computation can be based on paths.

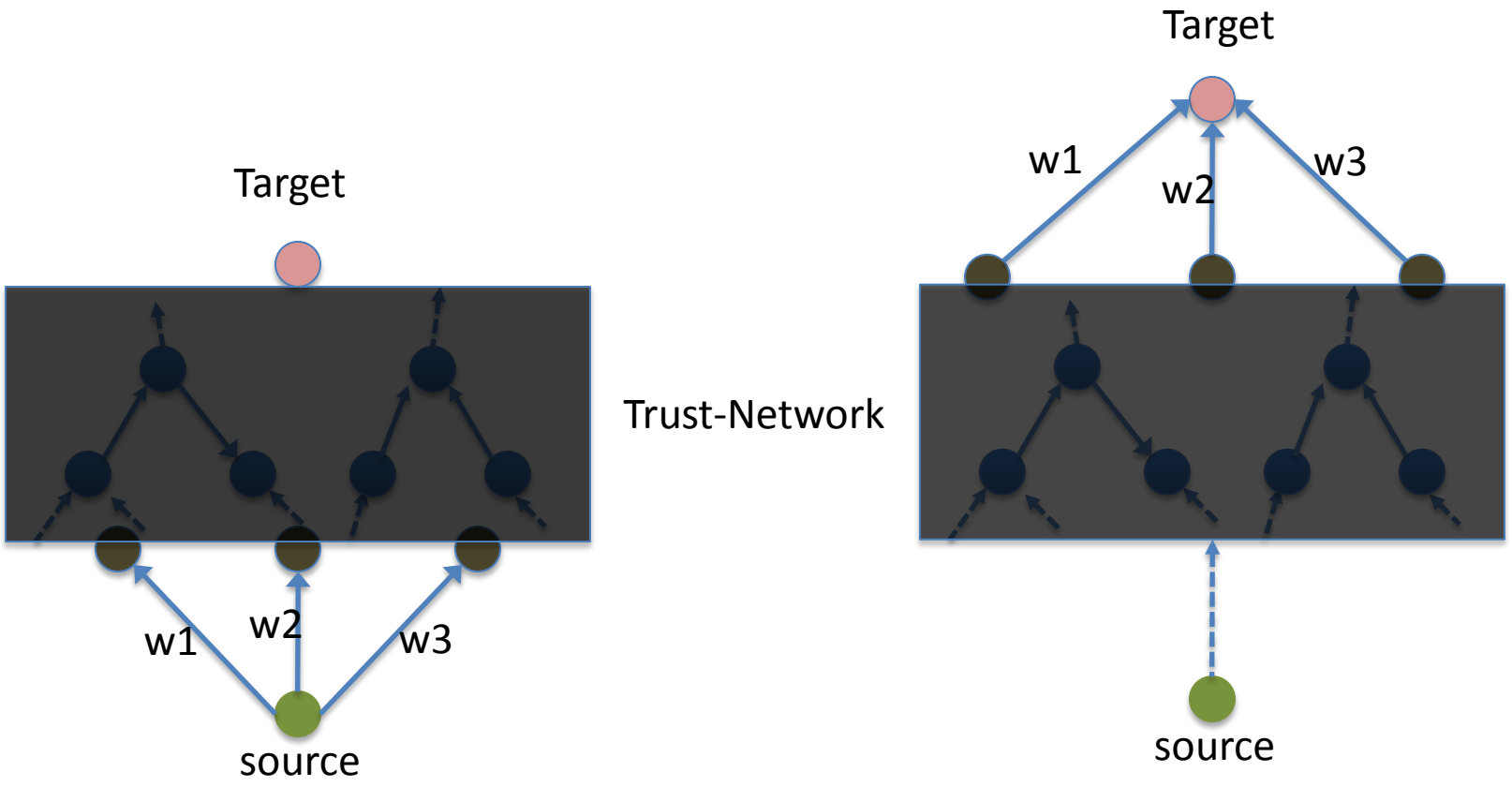
Golbeck – Hendler, 2006

Trust Propagation Algorithms

- Bottom-up
  - 1: Extract trust DAG (eliminate cycles)
  - 2: Predict trust score for a source in a target by aggregating trust scores in target inherited from **target's** "trusted" neighbors weighted with trust value in the corresponding neighbor.
    - Computation is level-by-level
    - Alternatively, computation can be based on paths.

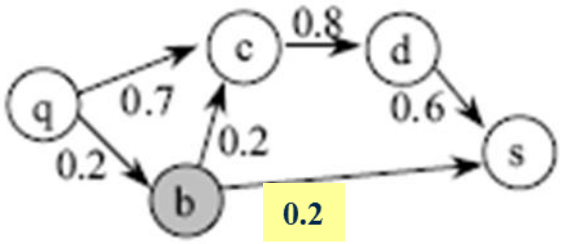
Massa-Avesani, 2005
Bintzios et al, 2006

Top-down vs Bottom-up (visualized)

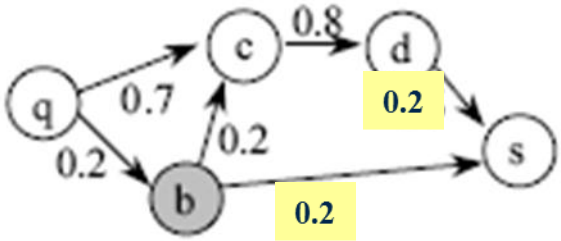


Example: Comparative Analysis

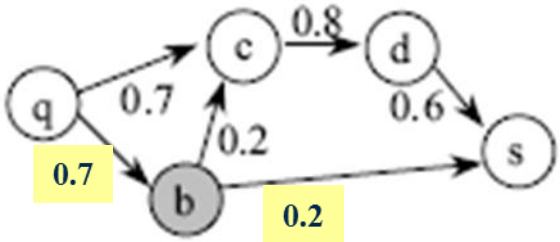
Same Interpretation:
q trusts s



Same Interpretation:
q distrusts s

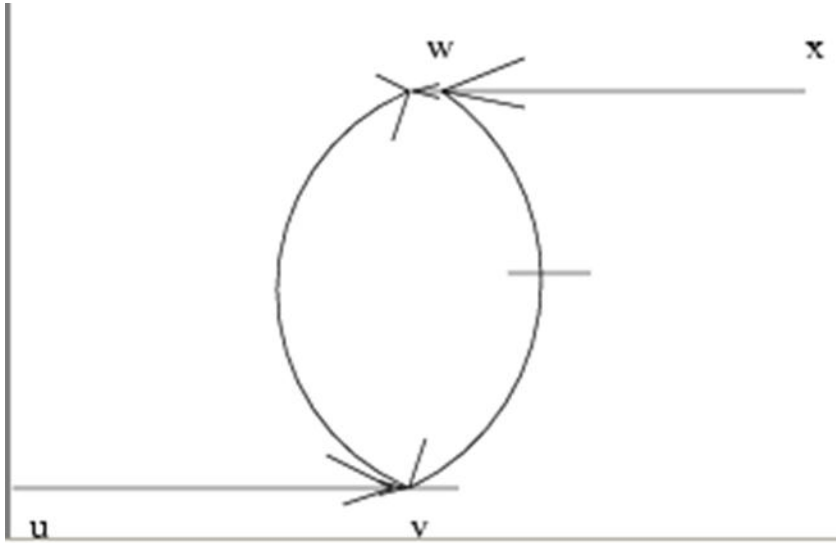


Different Interpretation:
q distrusts s (*Bintzios et al's*)
vs
q has no information about the trustworthiness of s (*our's, Golbeck rounding algorithm)*



Thirunarayan and Verma, 2007

Example: Well-founded Cyclic Trust Network



$T(u,v) = true$
$T(u,w) = true$
$T(v,w) = true$
$T(w,v) = false$
$T(x,w) = true$
$T(x,v) = false$
$T(\_,\_) = \bot$ Otherwise

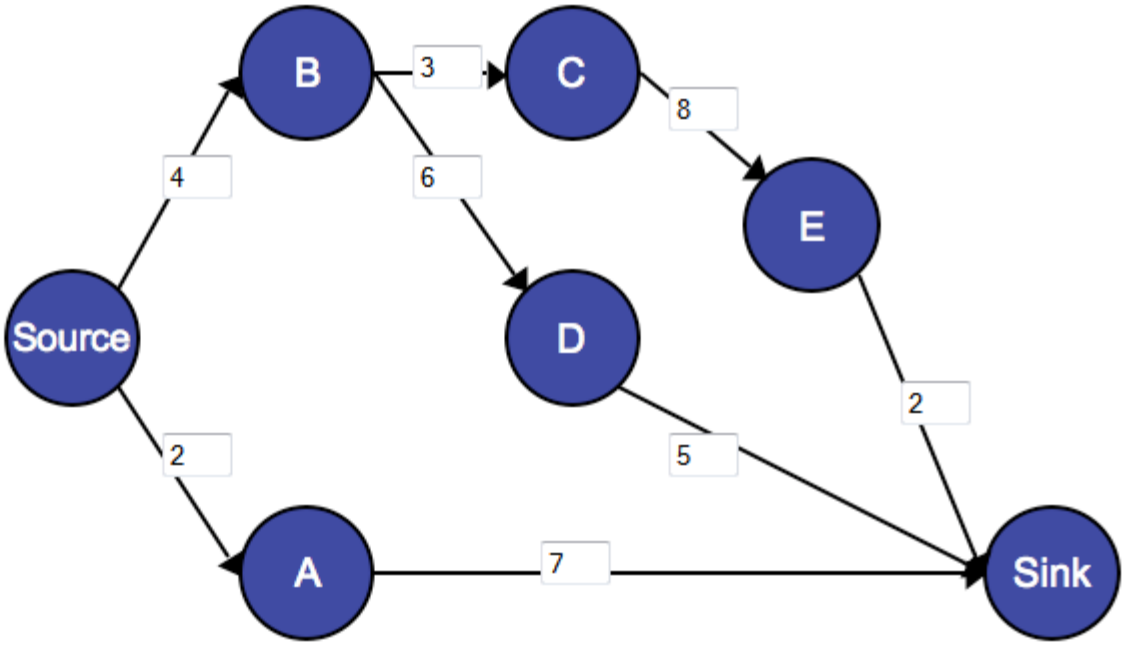
Thirunarayan and Verma, 2007

Example: Using TidalTrust Algorithm

Maximum Depth of Search: 4

Minimum Trust Value 1

Compute Inferred Trust Value 5



Indirect Trust : Referral + Functional
Variety of Bayesian Trust Models

With Applications to Mobile Ad hoc Networks
Wireless Sensor Networks, etc.

Direct Trust : Functional and Referral

- Trust link: {subject : agent, action}
- For MANETs (resp. cognitive radio)
  - Functional => Packet Forwarding (resp. quality of spectrum / channel)
  - Referral => Recommendations

Based on beta-reputation model:

Probability for trust = (S + 1) / (S + F +2)

where S = Number of good actions

F = Number of bad actions

Direct Trust : Functional and Referral

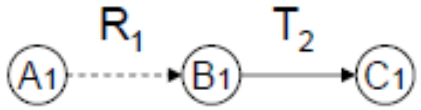
- Direct Trust for Packet Forwarding
  - S = Number of packets forwarded
  - F = Number of packets dropped
  - S + F = Total number of requests for packet forwarding
- Direct Trust for Recommendations
  - S = Number of times observed direct trust for packet forwarding ***approximates*** expected indirect trust for packet forwarding (trust over transit path : $r^{+}f$)
  - F = Number of times observed direct trust for packet forwarding ***does not approximate*** expected indirect trust for packet forwarding (trust over transit path : $r^{+}f$)

Indirect Trust : Functional and Referral

- Indirect Trust for Packet Forwarding
  - Used when direct trust is not available
    - » (overriding behavior)
  - Chain links for a path from a recommender to the target
    - Multiplicative
  - Aggregate over multiple (parallel) paths from recommenders to the target
    - Unclear, in general
- Indirect Trust for Recommendations
  - Obtained implicitly through computed referral trust

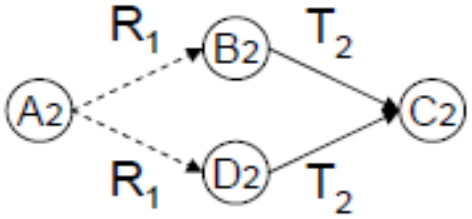
Trust Propagation Rules : Axioms for Trust Models

Rule 1: Concatenation propagation does not increase trust.



|T(A1,C1)| <= min(|R(A1,B1)|, |T(B1,C1)|)

Rule 2: Multipath propagation does not reduce trust.

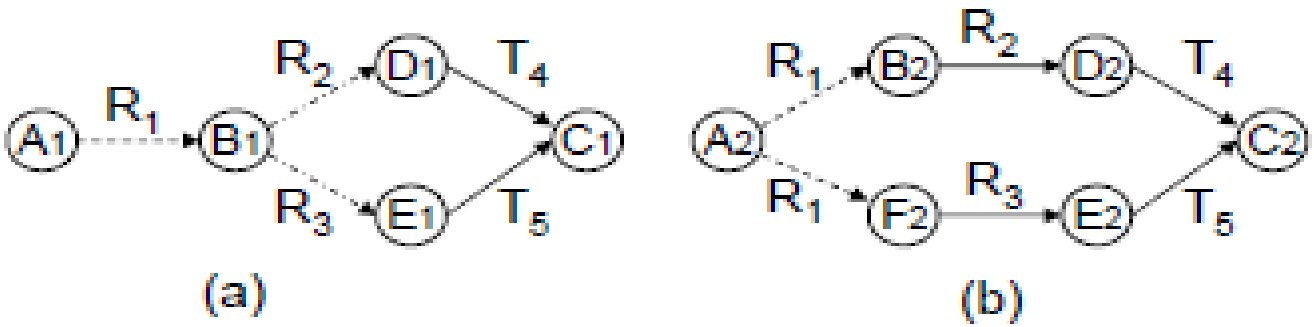


0<=T(A1,C1) <= T(A2,C2) for R1 > 0 and T2 >= 0
0>=T(A1,C1) >= T(A2,C2) for R1 > 0 and T2 < 0

Sun et al, 2006

(cont'd)

Rule 3: Trust based on multiple referrals from a single source should not be higher than that from independent sources.



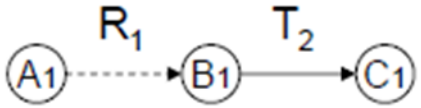
0<=T(A1,C1) <= T(A2,C2) for R1, R2, R3 > 0 and T2 >= 0
0>=T(A1,C1) >= T(A2,C2) for R1, R2, R3 > 0 and T2 < 0

Sun et al, 2006

Trust Propagation Rules : Implementation

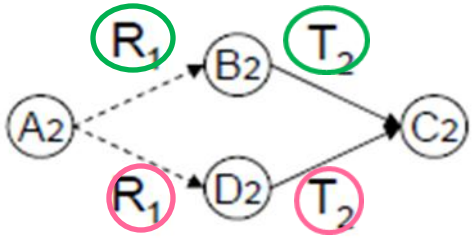
Rule 1: Concatenation propagation (reputation discounting)

$$T(A_1,C_1) = R_1 * T_2$$



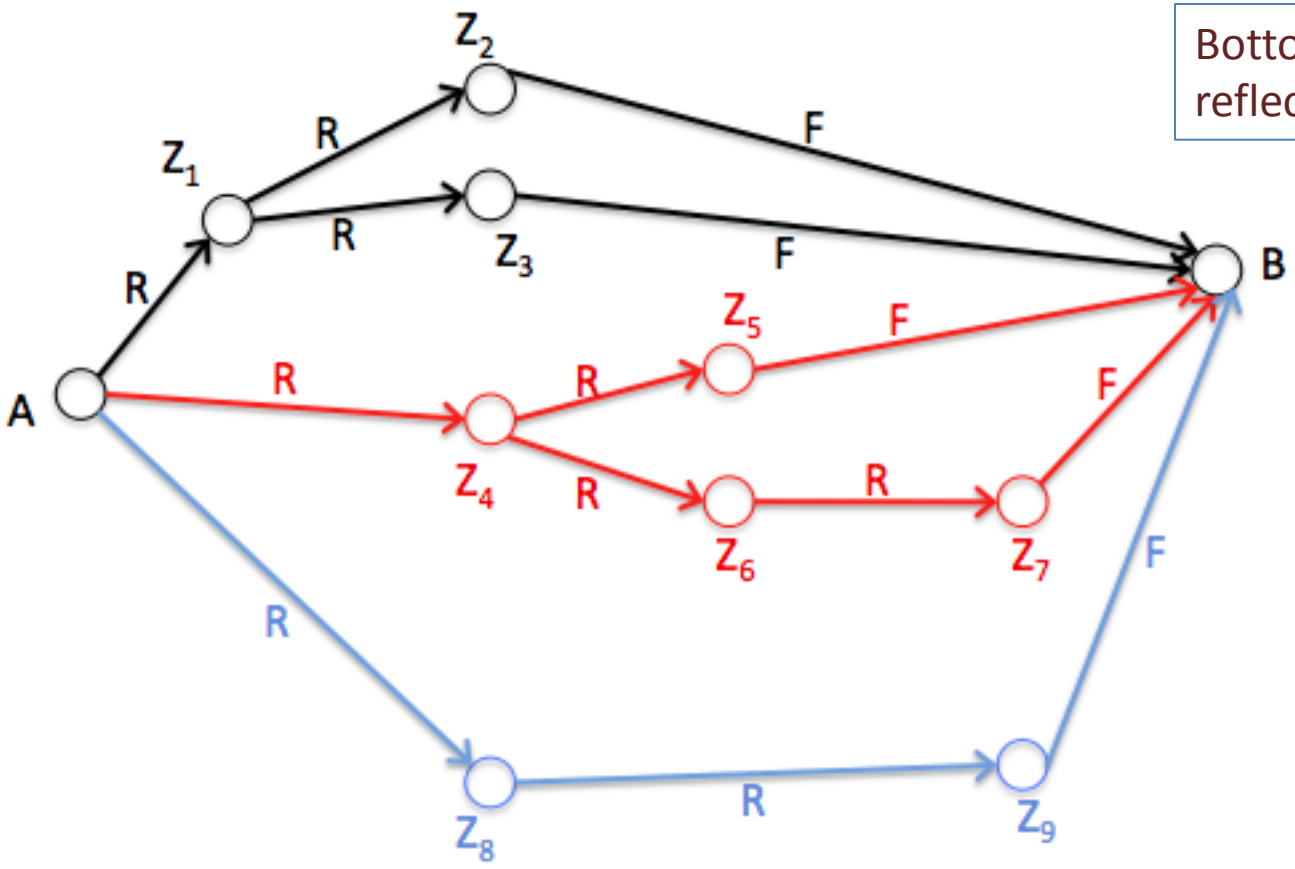
Rule 2: Multipath propagation (combining feedback)

$$T(A_2,C_2) = \frac{R_1(R_1*T_2)+R_1(R_1*T_2)}{R_1 + R_1}$$



Sun et al, 2006

Trust Paths Visualized for Scalability: Semantics unclear based on Sun et al's spec



Bottom-up computation reflects our needs better?

Trust : Functional and Referral

- Direct Trust for Primitive Actions based on
  - S = Number of success actions
  - F = Number of failed actions
  - S + F = Total number of actions
- Indirect Trust via Recommendations based on summing direct experiences of recommenders
  - S_k = Number of success actions for k^{th} recommender
  - F_k = Number of failed actions for k^{th} recommender
  - No chaining for referrals

Denko-Sun 2008

Cumulative Trust using Direct Experience and Recommendations

- Cumulative Trust is obtained by using total number of success actions and failed actions from direct experience (n_s,n_u) and from *i* (indirect experiences through) recommendations (n_s^r,n_u^r).

$$T_A(B) = \frac{n_s + n_s^r + 1}{(n_s + n_s^r + 1) + (n_u + n_u^r + 1)} = \frac{n_s + \sum_{k=1}^{i} n_s^k + 1}{n_s + n_u + \sum_{k=1}^{i} n_s^k + \sum_{j=1}^{i} n_u^j + 2}$$

Contents of [Ganeriwal et al, 2007] Paper

- (α,β)-parameters to compute trust of i in j is obtained by combining direct observations (α_j,β_j) with indirect observations (α_j^k,β_j^k) from k weighted by (α_k,β_k) using [Josang-Ismail, 2002] chaining/discounting rule.
  - Obtains cumulative trust by combining direct trust from a functional link and indirect trusts using paths containing one referral link and one functional link.
  - However, it does not distinguish functional and referral trust.

Trust Propagation Rules : Beta Reputation System

opinion = (belief **b**, disbelief **d**, uncertainty **u**)

in terms of (# correct values **r**, # errors **s**)

$$b = \frac{r}{r+s+2} \qquad d = \frac{s}{r+s+2} \qquad u = \frac{2}{r+s+2}$$

chaining of opinions

$$b = b1 * b2 \qquad d = b1 * d2$$

$$u = d1 + u1 + b1 * u2$$

Josang and Ismail, 2002

Trust Propagation Rules : Beta Reputation System

Rule 1: Concatenation propagation (reputation discounting)

$$r = 2*r1 * r2 \;/\; (s1+2)(r2+s2+2)+2*r1$$

$$s = 2*r1 * s2 \;/\; (s1+2)(r2+s2+2)+2*r1$$

Rule 2: Multipath propagation (combining feedback)

$$r = r1 + r2 \qquad s = s1 + s2$$

*Rule *:* Temporal Decay (Forgetting)

Josang and Ismail, 2002

Contents of [B-Trust, 2006] Paper

- Uses generic *K*-level discrete trust metric (as opposed to 2-level metric)
  - **E.g., (*very untrustworthy, untrustworthy, trustworthy, very trustworthy*)**
  - E.g., reminiscent of Amazon recommendation ratings
- Distributed (local), robust, lightweight, computational trust that takes into account context, subjectivity, and time
  - *a la* reputation-based approach
- *Application*: Pervasive computing

BUG -> FLAWED LEARNING?

- The approach does not clearly separate the use of stable background knowledge for applying Bayes' rule, from the need to dynamically learn background knowledge for gleaning trust from experience.
- As a result, the initial trust values ***do not*** change in response to experience.

Comparative Analysis

[Menko-T.Sun] : Beta-distribution based

- Direct functional trust and indirect functional trust (through direct referrals).
  - Trivial chaining.
  - One fixed context, local and distributed.
- Robustness improved by dropping extreme recommendations, though recommenders not distinguished.

Comparative Analysis

[Ganeriwal et al]: Beta-distribution based

- Functional and Referral trust mixed up.
  - Context glossed over, local and distributed.
- Robustness improved by chaining trust links of length 2, using Josang-Ismail opinion composition.
  - Recommenders distinguished.
  - Chaining *weighs* recommendations by recommender trust.

Comparative Analysis

[Y. Sun et al]: Beta-distribution based

- Functional and Referral trust separated.
  - One Context, hybrid (dynamically formed trust network) and distributed.
  - Information-theoretic approach
- Axiomatic specification and implementation of trust propagation (chaining and aggregation)
  - Incomplete w.r.t. arbitrary trust networks
- Robustness and Quality improved by analyzing dynamically formed trust network.

Comparative Analysis

[B-Trust et al]: Multi-valued trust - Bayesian

- Functional and Referral trust separated.
  - Context-based, local and distributed.
- Individualized aggregation, trivial chaining.
- Nice roadmap for theory, specification, and implementation of trust networks
  - Multi-valued Trust evolution : novel but buggy.

Comparative Analysis

[MLT-Approach]: Multi-level trust using Dirichlet Distribution

- Functional and Referral trust separated.
  - Context-based, local and distributed.
- Individualized aggregation, trivial chaining.
- Based on B-Trust roadmap but MLT evolution based on Dirichlet distribution: conceptually satisfactory and computationally efficient
- Example-based analysis for insights

Security Issues:
Threats and Vulnerabilities

Attacks and Robustness Analysis

Attacks

- Trust Management is an attractive target for malicious nodes.
  - Bad mouthing attack (Defamation)
    - Dishonest recommendations on good nodes (calling them bad)
  - Ballot stuffing attack (Collusion)
    - Dishonest recommendations on bad nodes (calling them good)
  - Sybil attack
    - Creating Fake Ids
  - Newcomer attack
    - Registering as new nodes

Attacks

- Inconsistency in time-domain
  - On-Off attack
    - Malicious node behaves good and bad alternatively to avoid detection
  - Sleeper attack
    - Malicious node acquires high trust by behaving good and then strikes by behaving bad
- Inconsistency in node-domain
  - Conflicting Behavior Attack
    - Provide one recommendation to one set of peers and a conflicting recommendation to a disjoint set of peers

Security : Robustness w.r.t Attacks

- Bad mouthing attack
  - *Example*: Competent nodes downplay competitions.
  - *Example*: Can diminish throughput due to lost capacity.
- Approach:
  - Separate functional and referral trust, updating referral trust to track good recommendations
  - Trust composition rules ensure that low or negative referral trust does not impact decision
    - Low trust nodes can be branded as malicious and avoided. (Not viable if majority collude.)

Security : Robustness w.r.t Attacks

- Ballot stuffing attack
  - *Example*: Malicious nodes collude to recommend each other.
  - *Example*: Can cause unexpected loss of throughput.
- Approach:
  - *Feedback* : Cross-check actual functional performance with expected behavior via referral, and update (reward/penalize) referral trust (in parent) accordingly (in addition to updating functional trust (in target))

Security : Robustness w.r.t. Attacks

- Sybil attack
  - Create Fake Ids to take blame for malicious behavior (dropping packets)
- Newcomer attack
  - Register as new node to erase past history
- Approach
  - Requires separate (key-based or security token-based) authentication mechanism (with TTP) to overcome these attacks.

Security : Robustness w.r.t Attacks

- On-Off attack
- Sleeper attack
  - *Example*: Due to malice or environmental changes
- Approach:
  - Use forgetting factor $(0<=\beta<=1)$:

    k good/bad actions at t1

    = k * $\beta^{(t2-t1)}$ good/bad actions at t2 (> t1)

Forgetting Factor

k good/bad actions at t1 = k * $\beta^{(t2-t1)}$ good/bad actions at t2 (> t1)

- High β value (0.9) enhances memorized time window, while low β value (0.001) reduces it.
  - High β enables *malicious* nodes (on-off/sleeper attackers) to use prior good actions to mask subsequent *intentional* bad actions.
    - Reduces reliability.
  - Low β forces *legitimate* nodes to be avoided due to short spurts of *unintentional* bad actions.
    - Reduces throughput.

Adaptive Forgetting Factor

- *Intuition*: Bad actions are remembered for a longer duration than good actions.
- Actions performed with high trust forgotten quicker than actions performed with low trust.

Choose β equal to (1 – p)

Choose β = 0.01 when p in [0.5,1] **else 0.9**

  - *Example*: Similar ideas used in Ushahidi
- *Note:* Effectively, more good actions are necessary to compensate for fewer bad actions, to recover trust.

Security : Robustness w.r.t. Attacks

- Conflicting Behavior Attack
  - Malicious node divide and conquer, by behaving differently (resp. by providing different recommendations) to different peers, causing peers to provide conflicting recommendations to source about the malicious node (resp. about some target), reducing source's referral trust in some peers.
    - Eventually, this causes recommendations of some peers to be ignored incorrectly.

Example

- Peer Node Set 1: 1, 2, 3, and 4
- Peer Node Set 2: 5, 6, 7, and 8
- Malicious node 0 behaves well towards nodes in Set 1 but behaves badly towards nodes in Set 2.
- When node 9 seeks recommendations from nodes in Set 1 U Set 2 on node 0, node 9 receives conflicting recommendations on malicious node 0, causing referral trust in nodes in Set 1 or nodes in Set 2 to be lowered.

  => Eventually throughput lowered

Security : Robustness w.r.t. Attacks

- Conflicting Behavior Attack
  - *Issue*: Can recommenders get feedback to reduce trust in malicious node? Otherwise, referral trust cannot be relied upon for detecting malicious nodes.

Security : Robustness w.r.t. Attacks

- If cumulative referral trust in B is computed using direct experiences of several recommenders,

$$(n_s^1, n_u^1), (n_s^2, n_u^2), \ldots, (n_s^l, n_u^l)$$

then it is possible to weed out extreme experiences using deviation from the mean trust value, where S is some chosen threshold.

$$|T_{cred}(B) - T_R(B)| > S$$

Denko-Sun 2008

Contents of [Ganeriwal et al, 2007] Paper

- Combined trust of i in j is obtained from direct observations (α_j,β_j) of j by i and indirect observations (α_j^k,β_j^k) from k to i weighted by (α_k,β_k) using [Josang-Ismail, 2002] chaining/weighting/discounting rule.
- This discounting rule makes the local trust computation resilient to bad mouthing $\alpha_j^k << \beta_j^k$ and ballot stuffing $\alpha_j^k >> \beta_j^k$ attacks from unreliable/malicious nodes $\alpha_k << \beta_k$.
- It requires aging to be resilient to sleeper attacks.

APPROACH/ METRIC	Trust Type / Context	Trust Model / Foundation	Robustness to Attacks
D[3] / Binary	Functional / One	Trivial chaining / Beta-PDF	Ballot-stuffing; Bad-mouthing
G[4] / Binary	Functional / Indistinguishable	Josang-Ismail discounting / Beta-PDF	Ballot-stuffing; Bad-mouthing; Sleeper and On-off
S[6] / Binary	Functional + Referral / One	Limited chaining and aggregation / Beta-PDF	Ballot-stuffing; Bad-mouthing; Sleeper and On-off
Q[28] / Multi-level	Functional + Referral / Multiple	No / Bayesian Ad Hoc	Ballot-stuffing; Bad-mouthing; Sleeper and On-off; Sybil
Ours / Multi-level	Functional + Referral / Multiple	No / Dirichlet-PDF	Ballot-stuffing; Bad-mouthing; Sleeper and On-off; Conflicting behavior

Research Challenges

(What-Why-How of trust?)

HARD PROBLEMS

Generic Directions

- Finding **online substitutes** for traditional cues to **derive measures of trust**.
- Creating **efficient** and **secure** systems for managing and deriving trust, in order to **support decision making**.

Josang et al, 2007

Robustness Issue

You can fool some of the people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time.

Abraham Lincoln,
16th president of US (1809 - 1865)

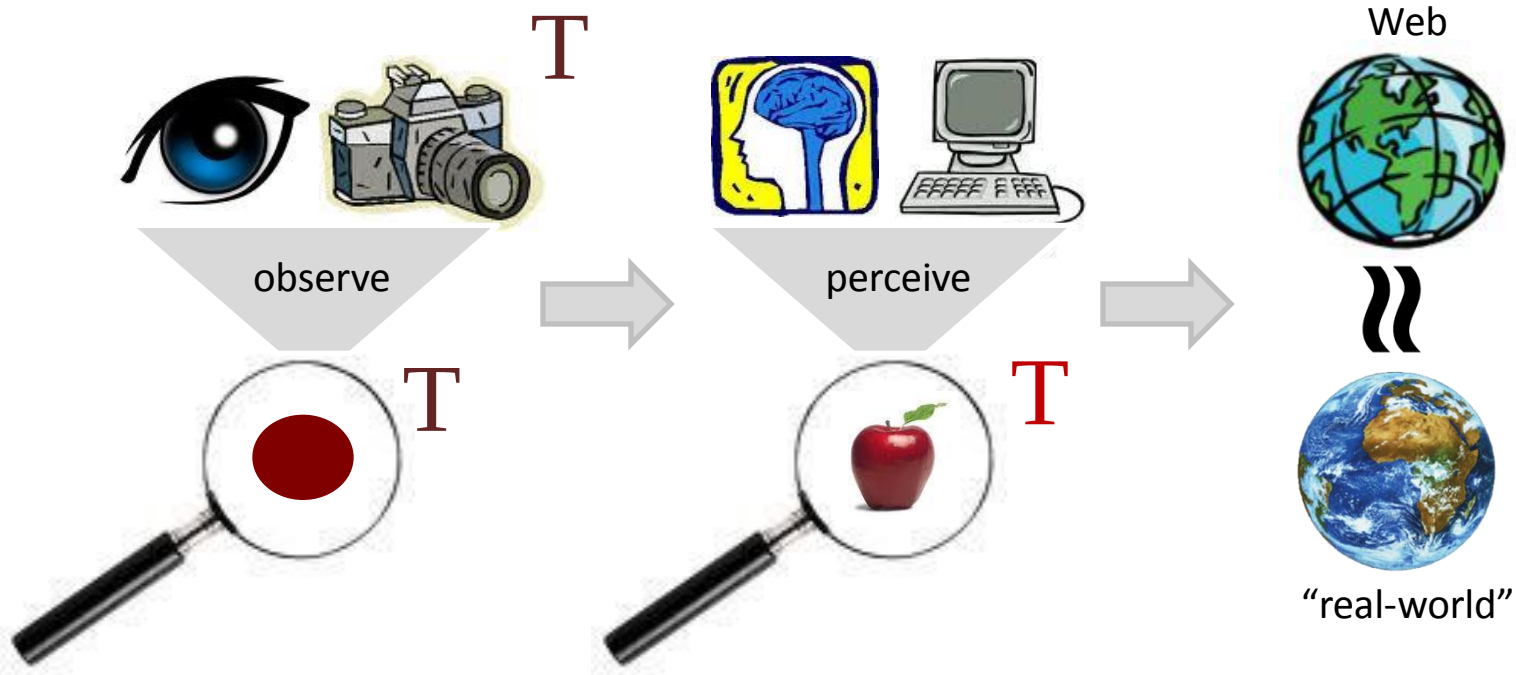
Trust : Social Networks *vs* Machine Networks

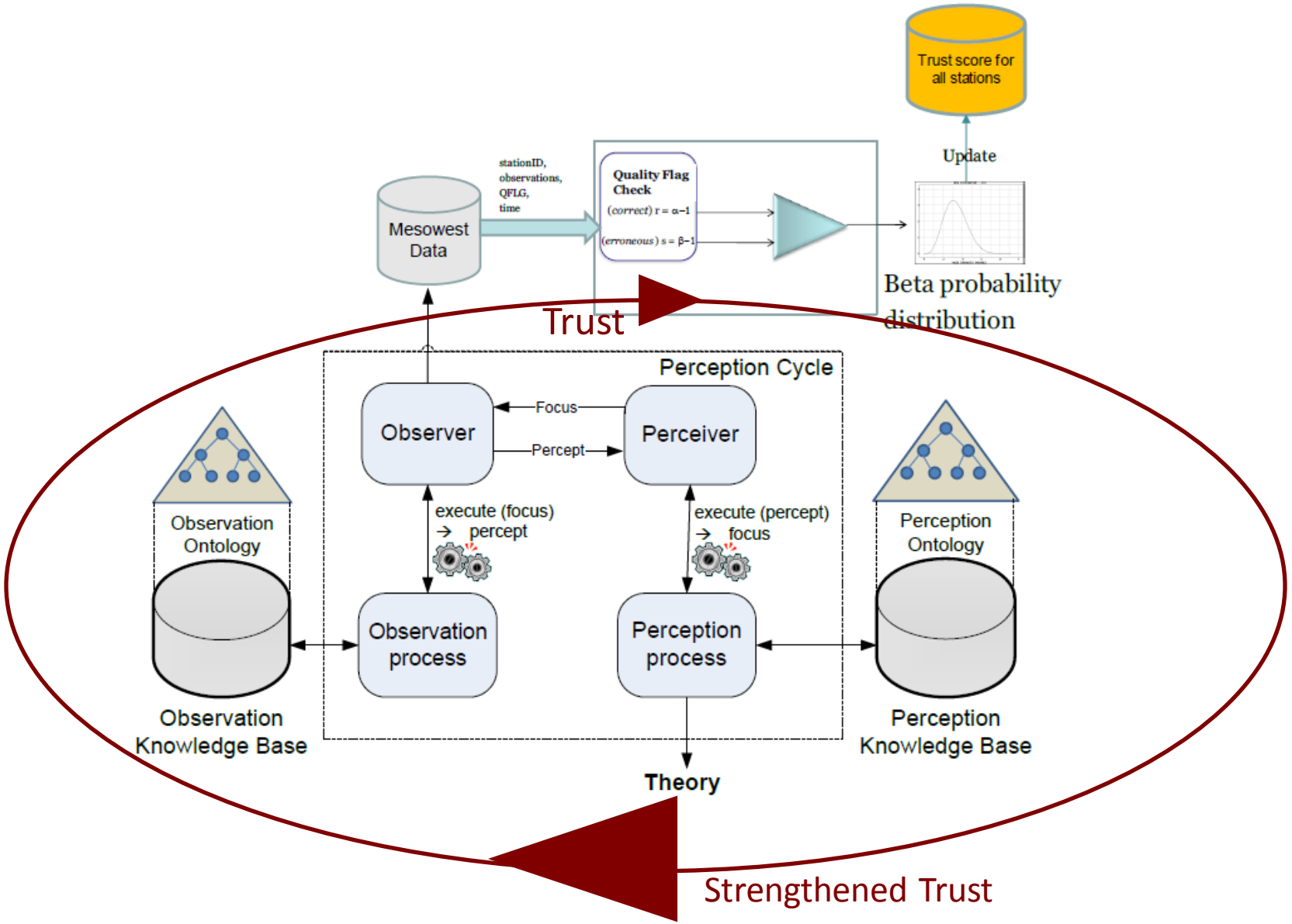
- In social networks such as Facebook, trust is often *subjective,* while in machine networks and social networks such as Twitter, trust can be given an *objective* basis and approximated by trustworthiness.
- *Reputation* is the perception that an agent creates through past actions about its intentions and norms.
  - Reputation can be a basis for trust.

Sensor Networks

Our Research

Abstract trustworthiness of sensors and observations to perceptions to obtain actionable situation awareness!



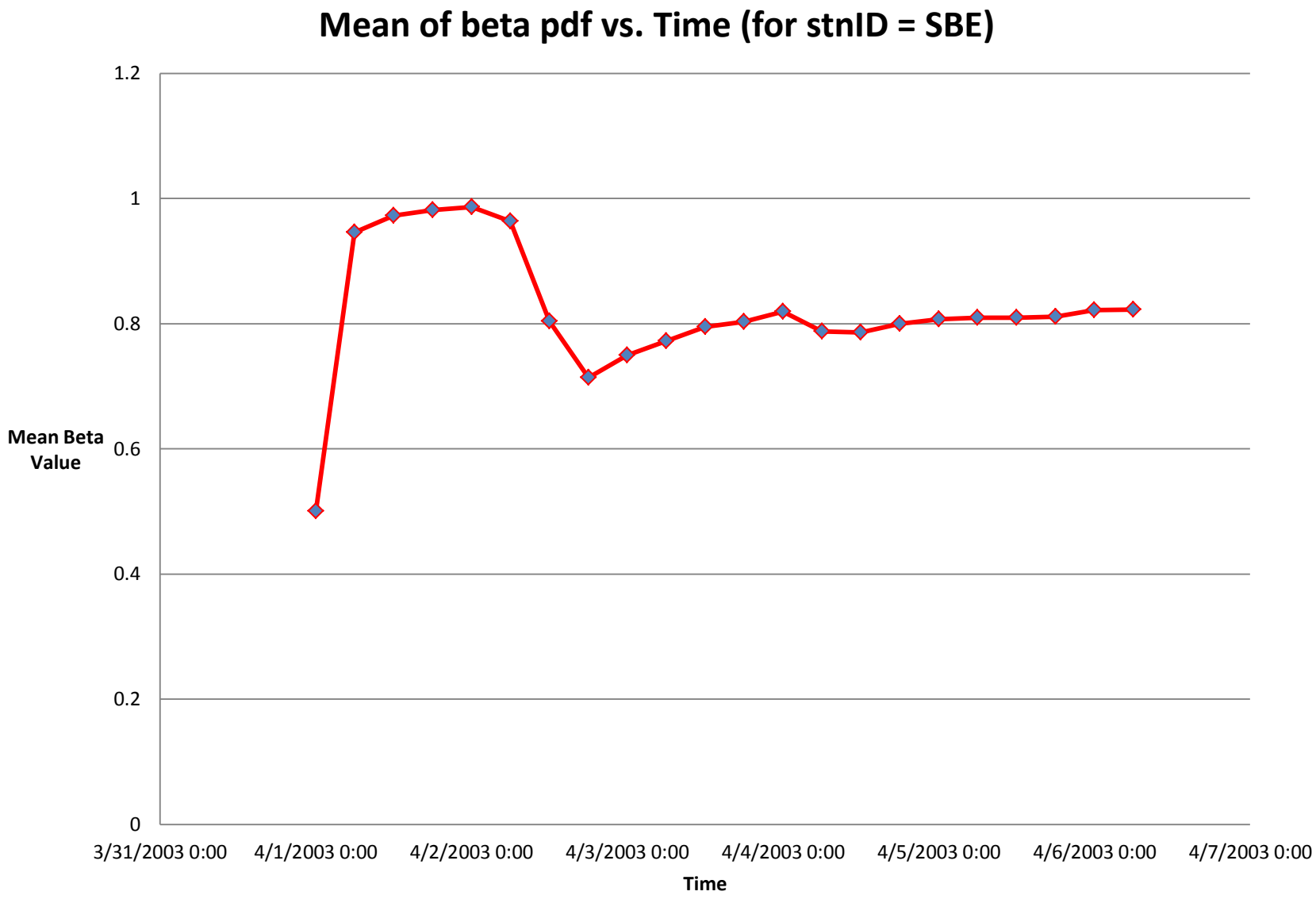

Trust score for all stations
Update
stationID, observations, QFLG, time
Mesowest Data
Quality Flag Check
(correct) r = α−1
(erroneous) s = β−1
Beta probability distribution
Trust
Perception Cycle
Observer
Focus
Percept
Perceiver
execute (focus) → percept
execute (percept) → focus
Observation Ontology
Perception Ontology
Observation process
Perception process
Observation Knowledge Base
Perception Knowledge Base
Theory
Strengthened Trust

Concrete Application

- Applied Beta-pdf to Mesowest Weather Data
  - Used quality flags (OK, CAUTION, SUSPECT) associated with observations from a sensor station over time to derive reputation of a sensor and trustworthiness of a perceptual theory that explains the observation.
  - Perception cycle used data from ~800 stations, collected for a blizzard during 4/1-6/03.

Concrete Application

- Perception Cycle
  - http://harp.cs.wright.edu/perception/
- Trusted Perception Cycle
  - http://www.youtube.com/watch?v=lTxzghCjGgU


Mean of beta pdf vs. Time (for stnID = SBE)
1.2
1
0.8
0.6
0.4
0.2
0
Mean Beta Value
3/31/2003 0:00
4/1/2003 0:00
4/2/2003 0:00
4/3/2003 0:00
4/4/2003 0:00
4/5/2003 0:00
4/6/2003 0:00
4/7/2003 0:00
Time

Research Issues

- Outlier Detection
  - Homogeneous Networks
    - Statistical Techniques
  - Heterogeneous Networks (sensor + social)
    - Domain Models
- Distinguishing between abnormal phenomenon (observation), malfunction (of a sensor), and compromised behavior (of a sensor)
  - Abnormal situations
  - Faulty behaviors
  - Malicious attacks

Ganeriwal et al, 2008

Social Networks

Our Research

- Study semantic issues relevant to trust
- Proposed model of trust/trust metrics to formalize *indirect* trust

Quote

- Guha et al:

  > While continuous-valued trusts are mathematically clean, from the standpoint of usability, most real-world systems will in fact use discrete values at which one user can rate another.

- E.g., Epinions, Ebay, Amazon, Facebook, etc all use small sets for (dis)trust/rating values.

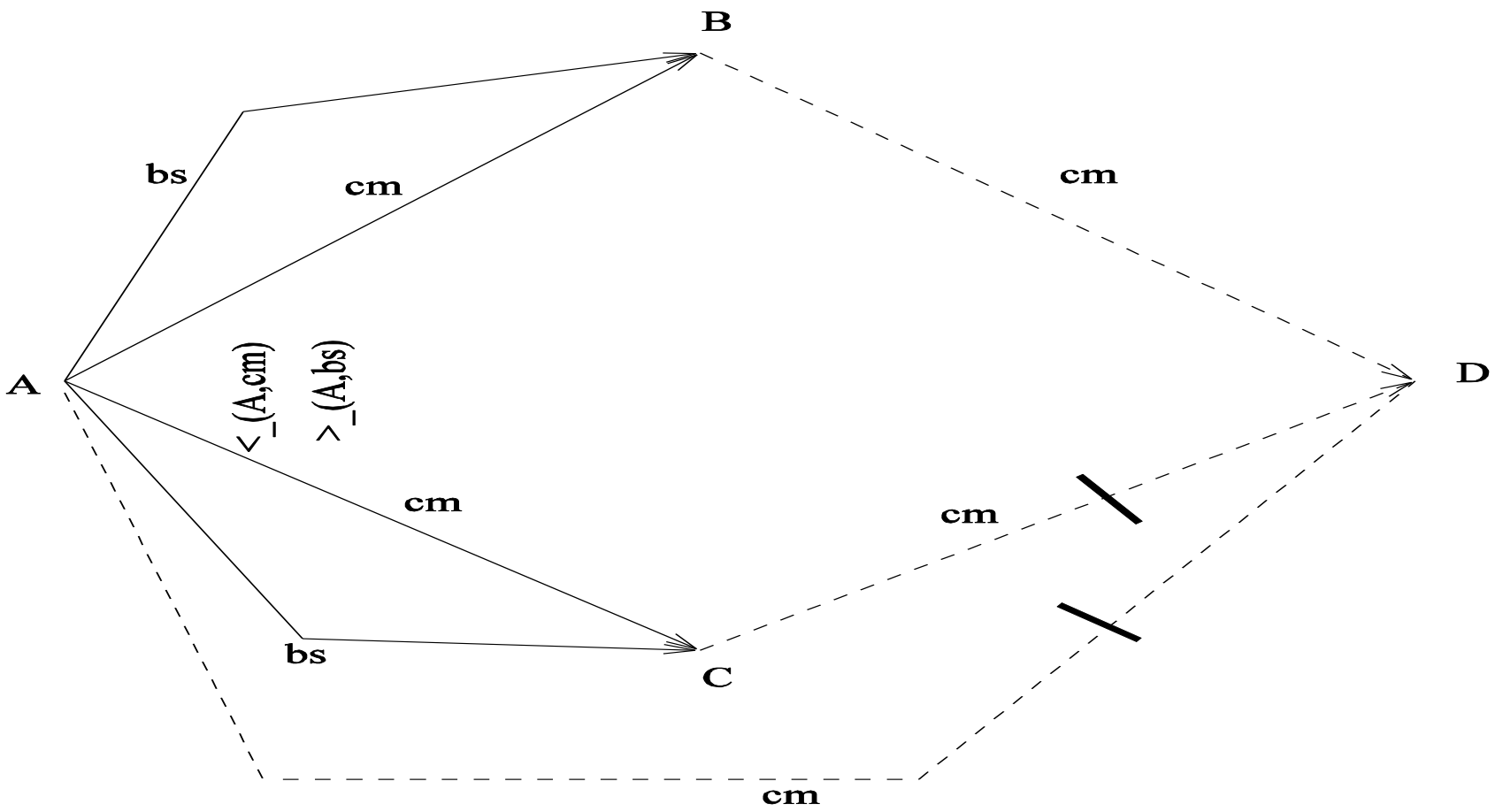
Our Approach

- Trust formalized in terms of partial orders (with emphasis on *relative* magnitude)
- *Local* but realistic semantics
  - Distinguishes *functional* and *referral* trust
  - Distinguishes *direct* and *inferred* trust
    - Direct trust *overrides* conflicting inferred trust
  - Represents *ambiguity* explicitly

Thirunarayan et al , 2009

Formalizing the Framework

- Given a trust network (Nodes AN, Edges RL U PFL U NFL with Trust Scopes TSF, Local Orderings $\preceq_{ANxAN}$), specify when a source can trust, distrust, or be ambiguous about a target, reflecting local semantics of:
  - *Functional* and *referral* trust links
  - *Direct* and *inferred* trust
  - *Locality*



Referral trust link (In recommendations)

Functional trust link (For capacity to act)

Nonfunctional trust link (For lack of capacity to act)

Evidence in support of Referral Trust: a_i can referral trust a_j in trust scope ts if there is an explicit trust link from a_i to a_j, or there is a successor a_k of a_i that referral trusts a_j in trust scope ts.

$$\forall a_i, a_j \in AN : a_i \textbf{ can referral trust } a_j \textbf{ in trust scope } ts \text{ if}$$

$$[\,(a_i, a_j) \in \mathrm{RL} \;\wedge\; ts \in TSF(a_i, a_j)\,] \;\vee$$

$$[\exists a_k \in \mathrm{AN} : \;\; (a_i, a_k) \in \mathrm{RL} \;\wedge\; ts \in TSF(a_i, a_k) \;\wedge\; ts \in \mathcal{R}(a_k, a_j)\,]$$

$$\| \text{ Undefeated functional trust of } a_i \textbf{ in } a_j \textbf{ for } ts \| =$$

$$\| \{(a_i, a_k) \in \text{RL} \mid ts \in TSF(a_i, a_k) \wedge (ts, true) \in \mathcal{F}(a_k, a_j)$$

$$\wedge \neg \exists a_l \in \text{AN} : (a_k \prec_{(a_i, ts)} a_l)$$

$$\wedge (a_i, a_l) \in \text{RL} \wedge ts \in TSF(a_i, a_k) \wedge (ts, bf) \in \mathcal{F}(a_l, a_j)\} \wedge bf \geq false \|$$

and

$$\| \text{ Undefeated nonfunctional trust of } a_i \textbf{ in } a_j \textbf{ for } ts \| =$$

$$\| \{(a_i, a_k) \in \text{RL} \mid ts \in TSF(a_i, a_k) \wedge (ts, false) \in \mathcal{F}(a_k, a_j)$$

$$\wedge \neg \exists a_l \in \text{AN} : (a_k \prec_{(a_i, ts)} a_l)$$

$$\wedge (a_i, a_l) \in \text{RL} \wedge ts \in TSF(a_i, a_k) \wedge (ts, bt) \in \mathcal{F}(a_l, a_j)\} \wedge bt \geq true \|$$

Evidence in support of Positive Functional Trust: a_i can functional trust a_j in trust scope ts if there is an explicit positive functional trust link from a_i to a_j, or there is majority of most referral trusted successors a_k of a_i that functional trust a_j rather than distrust a_j. In other words, for the purposes of a_j in trust scope ts, there are more endorsements than disapprovals via a_i's successors. We introduce a factor Kp to quantify the strength of majority for positive functional trust. Normally, its value is at least 1, and for simple majority, Kp is equal to 1.

$$\forall a_i, a_j \in AN : a_i \textbf{ can functional trust } a_j \textbf{ in trust scope } ts \text{ if}$$

$$(a_i, a_j) \in \text{PFL} \ \wedge \ ts \in TSF(a_i, a_j) \ \vee$$

$$\frac{\| \text{ Undefeated functional trust of } a_i \textbf{ in } a_j \textbf{ for } ts \ \|}{\| \text{ Undefeated nonfunctional trust of } a_i \textbf{ in } a_j \textbf{ for } ts \ \|} > Kp$$

Similarly for Evidence in support of Negative Functional Trust.

Benefits of Formal Analysis

- Enables detecting and avoiding unintended consequences.
  - An earlier formalization gave priority to "*certain*" conclusion from less trustworthy source over "*ambiguous*" conclusion from more trustworthy source.

The whole problem with the world is that fools and fanatics are always so certain of themselves, but wiser people so full of doubts. — Betrand Russell

Practical Issues

- Refinement of numeric ratings using reviews in product rating networks
  - Relevance : Separate ratings of vendor or about extraneous features from ratings of product
    - E.g., Issues about Amazon's policies
    - E.g., Publishing under multiple titles (Paul Davies' "The Goldilock's Enigma" vs. "Cosmic Jackpot")
  - Polarity/Degree of support: Check consistency between rating and review using sentiment analysis; amplify hidden sentiments
    - E.g., rate a phone as 1-star because it is the best ☹

Research Issues

- Determination of trust / influence from social networks
  - Text analytics on communication
  - Analysis of network topology
    - E.g., follower relationship, friend relationship, etc.
- Determination of untrustworthy and anti-social elements in social networks
- HOLY GRAIL: Direct Semantics in favor of Indirect Translations

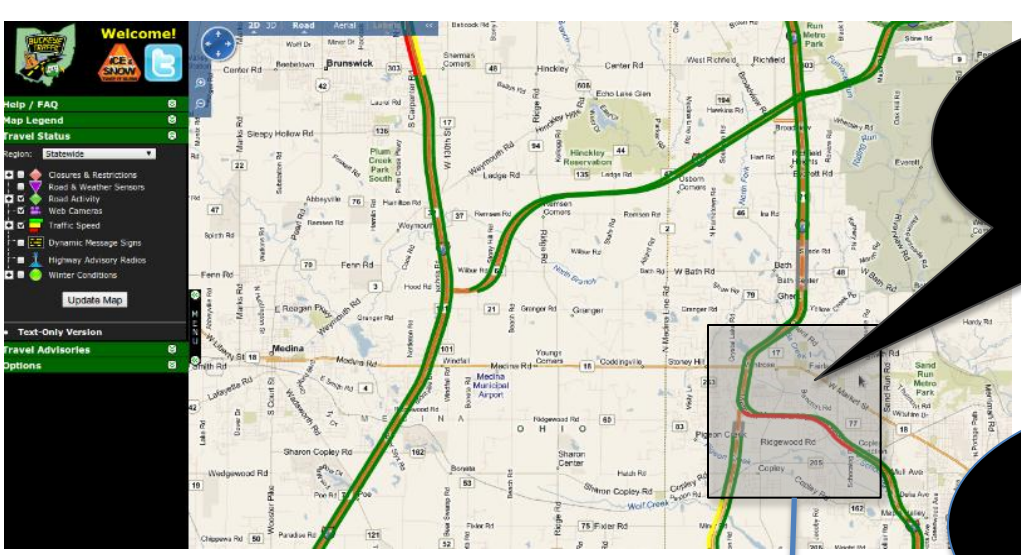
Research Issues

- Improving Security : Robustness to Attack
  - How to exploit different trust processes to detect and recover from attacks?
    - Bad mouthing attack
    - Ballot stuffing attack
    - Sleeper attack
      - Temporal trust discounting proportional to trust value
      - Using *policy-based process* to ward-off attack using *reputation-based process*
    - Sybil attack
    - Newcomer attack

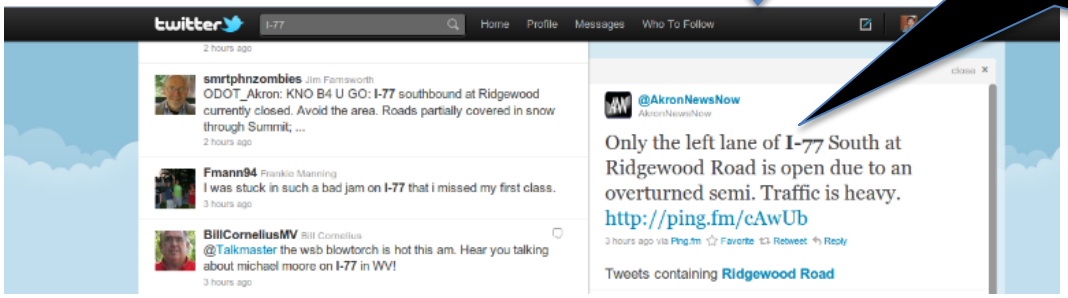
Research Issues

- Intelligent integration of mobile sensor and social data for situational awareness
  - To exploit complementary and corroborative evidence provided by them
  - To obtain qualitative and quantitative context
  - To improve robustness and completeness
  - To incorporate socio-cultural, linguistic and behavioral knowledge as part of ontologies to improve semantic processing and analysis of data

Complementary and Corroborative Information



Sensors observe slow moving traffic



Complementary information from social networks

Corroborative Evidence

Interpersonal and Ecommerce Networks

Research Issues

- Linguistic clues that betray trustworthiness
- Experiments for gauging interpersonal trust in real world situations
  - *Techniques and tools to detect and amplify useful signals in Self to more accurately predict trust and trustworthiness in Others

*IARPA-TRUST program

Research Issues

- Other clues for gleaning trustworthiness
  - Face (in photo) can effect perceived trustworthiness and decision making
  - Trust-inducing features of e-commerce sites can impact buyers
  - Personal traits: religious beliefs, age, gullibility, benevolence, etc
  - Nature of dyadic relationship

Research Issues

- Study of cross-cultural differences in trustworthiness qualities and trust thresholds to better understand
  - Influence
    - What aspects improve influence?
  - Manipulation
    - What aspects flag manipulation?

Collaborative Systems : Grid and P2P Computing

Research Issues

- Trust-aware resource management and scheduling
  - Clients specify resource preferences/requirements/constraints
- Trust models for P2P systems
  - To detect bad domains
  - To detect bogus recommendations and attacks

Azzedin and Maheshwaran, 2002-2003
Azzedin and Ridha, 2010
Bessis et al, 2011

Bayesian Trust Management Framework : Multi-level Trust Metric

Illustrating a General Approach

Quercia et al 2006
Josang and Haller 2007
Thirunarayan et al 2012

Outline

- *Motivation* : Multi-level trust management
- *Mathematical Foundation*: Dirichlet Distribution
- *Implementation and Behavior Details*:
  - Local Trust Data Structures
  - Trust Formation
  - Bayesian Trust Evolution
- *Analysis of Robustness to Attacks*: Security
- *Evaluation*: Example trace vs. experiment

Motivation

- Uses K-level discrete trust metric
  - E.g., Amazon's 5-star trust metric can be interpreted as signifying (very untrustworthy, untrustworthy, neutral, trustworthy, very trustworthy) or (very dissatisfied, dissatisfied, neutral, satisfied, very satisfied).

Approach

- Multi-level trust management approach formalizes a *distributed*, *robust*, *lightweight*, *computational trust* that takes into account *context*, *subjectivity*, and *time*.
- Applies Dirichlet distribution, a generalization of Beta-distribution.

Dirichlet Distribution

K-level Trust Metric

- K-level trust probability vector:

$$\mathbf{x = (x_1, \ldots, x_K)}$$

where $(x_1 + \ldots + x_K = 1)$.

- *Example*: If a 5-star rating system has 50 people giving 5-stars, 20 people giving 4-stars, 5 people giving 3-stars, 5 people giving 2-stars, and 20 people giving 1-star, then the 5-level trust metric probability vector is (0.5,0.2,0.05,0.05,0.2).

Trust and Experience

- Experience is a realization of latent trust and helps predicting trust.
- Probability of an experience-level sequence,

with α_1 - 1 counts of level 1 experience, …, α_K - 1 counts of level K experience is:

$$\prod_{i=1}^{K} x_i^{\alpha_i - 1} \quad * \left((\alpha_1 + \ldots + \alpha_K - K) \,! \,/ \, (\alpha_1 - 1 \,! * \ldots * \alpha_K - 1 \,!) \right)$$

Dirichlet Distribution

- The Dirichlet distribution is the probability density function for $x = (x_1, \ldots, x_K)$ given $(\alpha_1,\ldots,\alpha_K)$:

$$f(x_1,\ldots,x_{K-1};\alpha_1,\ldots,\alpha_K) = \frac{1}{\mathrm{B}(\alpha)}\prod_{i=1}^{K} x_i^{\alpha_i-1}$$

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{K}\Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}, \qquad \alpha = (\alpha_1,\ldots,\alpha_K).$$

$$\Gamma(n) = (n-1)!$$

Why use Dirichlet Distribution?

- If the prior distribution of x is uniform, then the Dirichlet family of distribution shown below gives posterior distribution of x after α_i-1 occurrences of level i experience with probability x_i, for each i in [1, K]:

$$f(x_1, \ldots, x_{K-1}; \alpha_1, \ldots, \alpha_K)$$

Why use Dirichlet Distribution?

- *Dirichlet distribution is a conjugate prior for multinomial distribution.*
- *Consequence*:
  - Estimated distribution updated for a new experience at level i, by just incrementing α_i parameter.
  - In contrast: if prior distribution is different from Dirichlet, then it is *conceptually hard* to comprehend and *computationally inefficient* to compute posterior distribution, in general.
  - Icing on the cake: Uniform distribution (signifying ignorance) is Dirichlet!

Dirichlet distribution is a *conjugate prior* for multinomial distribution.

$$prob(x|c) = \frac{prob(c|x)prob(x)}{prob(c)}$$

$$f(x|c) \sim Multinomial(c|x) * Dirichlet(x|\alpha)$$

$$f(x|c) \sim \prod_{i=1}^{K} x^{c_i} * \prod_{i=1}^{K} x^{(\alpha_i - 1)}$$

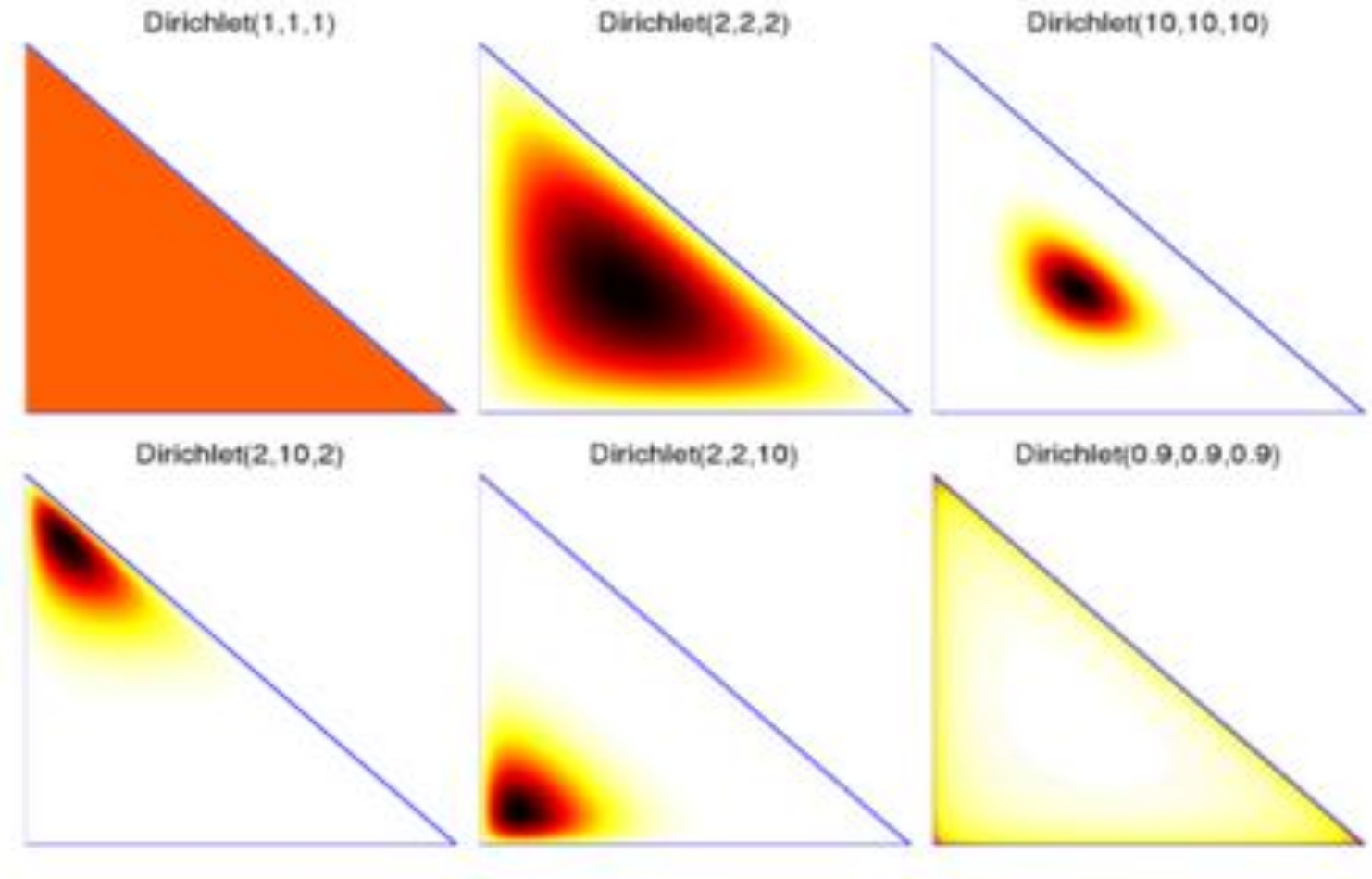
$$f(x|c) \sim \prod_{i=1}^{K} x^{(\alpha_i + c_i - 1)}$$

$$f(x|c) \sim Dirichlet(x|\alpha + c)$$

Why use Dirichlet Distribution?

- Convenient Abstraction
  - Abstraction of K-level Dirichlet distribution by combining different levels still yields Dirichlet distribution with the corresponding parameters merged.
    - Conceptually and computationally pleasing property

Visualizing Dirichlet Distribution (K=3): Color Density plot on 2D simplex



Dynamic Trustworthiness

- Best estimate of trust for Dir(α_1,…,α_K) (gleaned from (α_i-1) experiences at level i, for all i in [1,K]) is the mean vector (α_1/α_0,…,α_K/α_0), and the associated confidence is the variance vector.

Define $\alpha_0 = \sum_{i=1}^{K} \alpha_i$.

$$\mathrm{E}[X_i] = \frac{\alpha_i}{\alpha_0},$$

$$\mathrm{Var}[X_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} = \frac{\mathrm{E}[X_i](1 - \mathrm{E}[X_i])}{(\alpha_0 + 1)}.$$

Implementation and Behavior Details

Local Data Structures

- To store relevant information to compute direct (functional) and indirect (referral) trust.
- Each node maintains locally, for each peer and each context, four vectors of length K.

Local Data Structures

- Direct Trust Vector: Peers X Contexts X Peers -> Probability-Vector-K
- $dtv(px,c,py) = (d_1,d_2,\ldots,d_K)$

- Direct Experience Matrix: Peers X Contexts X Peers -> Count-Vector-K
- $dem(px,c,py) = (ec_1,\ldots,ec_K)$

Local Data Structures

- Recommended Trust Vector: Peers X Contexts X Peers -> Probability-Vector-K
- $rtv(px,c,py) = (r_1,r_2,\ldots,r_K)$
- Sent Recommendation Matrix: Peers X Contexts X Peers -> Count-Vector-K
- $srm(px,c,py) = (sr_1,\ldots,sr_K)$

Local Data Structures

- *Initialization:* To reflect complete ignorance via uniform distribution, we set the probability vectors dtv and rtv to (1/K,…,1/K), and the elements of the count vector dem and srm to (0,…,0).
- These are Dirichlet distributed in the limiting case where α_i's are 1.

Trust Formation

- Overall trust vector is *weighted* combination of direct trust vector and recommended trust vector.
- Weights determined using
  - Objective confidence values using variance (deviation from the mean)
  - Subjective relative preference for direct experience over recommendations
    - Dependence on recommended trust yet to be explored

Trust Decision

- Assuming that trust-level scale is *linear*, the trust distribution vector $(d_1,d_2,\ldots,d_K)$ can be mapped to the closed interval [0,1], or to consolidated trust level, in order to act.

- Trust threshold should be determined based on the context and risk tolerance / disposition / propensity to trust.

Trust Evolution

- Direct (recommended) trust vectors are updated for a new experience (recommendation).
- *Key Idea*: Dirichlet distribution is the conjugate prior of the multinomial distribution. So it is adequate to maintain counts of direct experience and sent recommendations, to best estimate direct trust and recommended trust vectors respectively.

Trust Evolution

- Simple Scheme (Direct Trust)

For a new experience at level i,

$dem(px,x,py) = (ec_1,\ldots,ec_K)$ *becomes*

$dem^{new}(px,x,py) = (ec_1,\ldots, \mathbf{ec_i+1},\ldots,ec_K)$

and $dtv(px,c,py)$ *becomes*

$dtv^{new}(px,c,py) = (d_1,d_2,\ldots,d_K)$

where $\mathbf{di = ec_i+1 \,/\, (ec_1 + \ldots + ec_k+1)}$ *and*

$\mathbf{dj = ec_j \,/\, (ec_1 + \ldots + ec_k+1)}$

for each j in [1,K] and j =/= i.

Trust Evolution

- Robust Scheme

To incorporate differential aging of experience counts as a function of their level (and to incorporate "long term memory for low-level experience and short term memory for high-level experience"), we use a decay vector $(\lambda_1,\ldots,\lambda_K)$, where $1 >= \lambda_1 >= \ldots >= \lambda_K > 0$, that modifies update rule as:

Trust Evolution

- Robust Scheme (Direct Trust)

For a new experience at level i,

$dem(px,x,py) = (ec_1,\ldots,ec_K)$ *becomes*

$dem^{new}(px,x,py) = (ec_1,\ldots, \mathbf{ec_i + 1},\ldots,ec_K)$.

For every clock tick (with context-based delay),

$dem(px,x,py) = (ec_1,\ldots,ec_K)$ *becomes*

$\mathbf{dem^{new}(px,x,py) = (\lambda_1 * ec_1,\ldots, \lambda_K * ec_K)}$

Trust Evolution

- Robust Scheme (Direct Trust)

For every clock unit and new experience,

dtv(px,c,py) becomes

$dtv^{new}(px,c,py) = (d_1,d_2,\ldots,d_K)$

where $d_i = ec_i / (ec_1 + \ldots + ec_k)$

for each i in [1,K].

- *Subtlety*: Experience counts should *saturate at 1* rather than diminish to 0 with time. (See code)

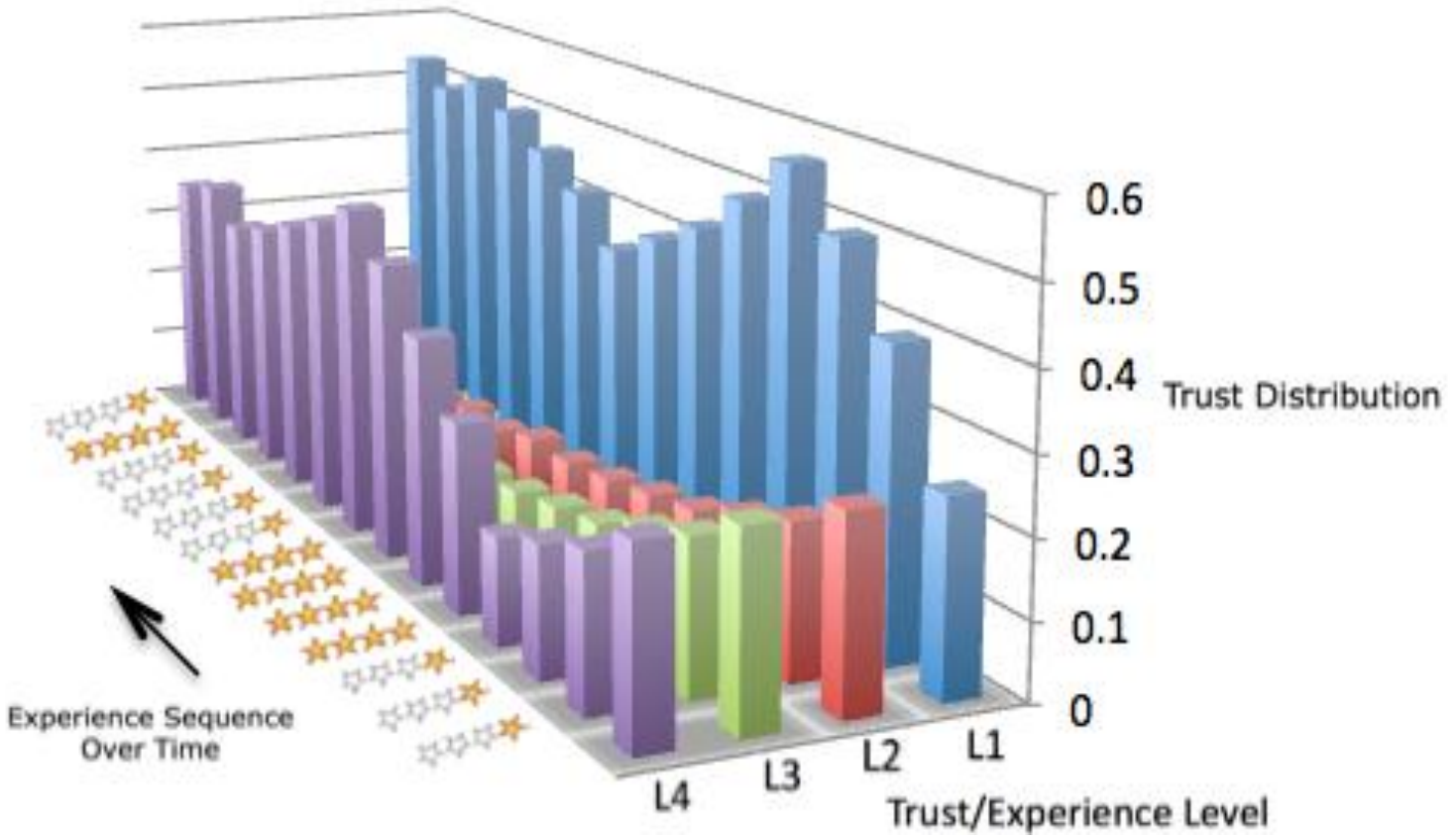
Trust Evolution Illustrated

Experience Sequence	Final Trust Distribution (Simple Scheme)	Final Trust Distribution (Robust Scheme)
[1,1,1]	(0.57,0.14,0.14,0.14)	(0.55,0.15,0.15,0.15)
[1,4,1,4]	**(0.375,0.125,0.125,0.375)**	**(0.42,0.14,0.14,0.29)**
[1,1,4,4,4,4,1,1]	(0.42,0.08,0.08,0.42)	(0.5,0.1,0.1,0.3)
[1,1,4,4,4,4,1,1,1]	(0.53,0.07,0.07,0.33)	(0.64,0.1,0.1,0.17)
[2,3,2,3]	**(0.125,0.375,0.375,0.125)**	**(0.16,0.4,0.3,0.14)**

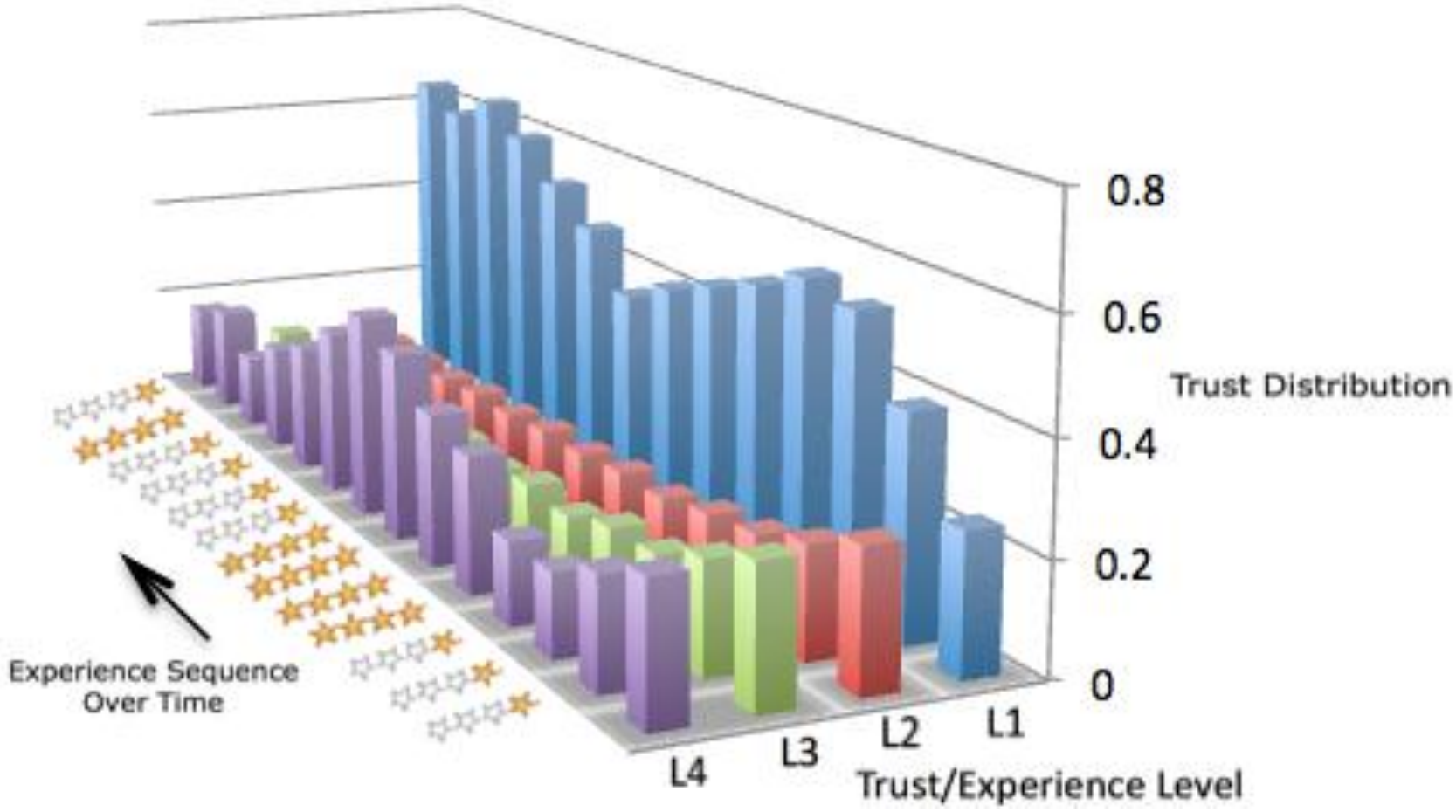
Trust Evolution Illustrated

Experience Sequence Value	Trust Distribution Trace (Simple Scheme)	Trust Distribution Trace (Robust Scheme)	Beta-pdf (cf. n =2)
	(0.25,0.25,0.25,0.25)	(0.25,0.25,0.25,0.25)	**0.5**
1	(0.4,0.2,0.2,0.2)	(0.4,0.2,0.2,0.2)	0.33
1	(0.5,0.17,0.17,0.17)	(0.53,0.165,0.155,0.15)	0.25
1	**(0.57,0.14,0.14,0.14)**	**(0.55,0.15,0.15,0.15)**	0.2
n	(0.5,0.125,0.125,0.25)	(0.5,0.13,0.12,0.25)	0.33
n	(0.44,0.11,0.11,0.33)	(0.46,0.135,0.135,0.27)	0.43
n	**(0.4,0.1,0.1,0.4)**	**(0.42,0.12,0.11,0.35)**	**0.5**
n	(0.36,0.1,0.1,0.45)	(0.37,0.12,0.12,0.38)	0.55
1	(0.42,0.08,0.08,0.41)	(0.47,0.11,0.11,0.31)	**0.5**
1	**(0.46,0.08,0.08,0.38)**	**(0.53,0.11,0.11,0.24)**	0.45
1	**(0.5,0.07,0.07,0.35)**	**(0.6,0.1,0.1,0.2)**	0.41
1	(0.53,0.07,0.07,0.33)	(0.65,0.1,0.1,0.14)	0.38
n	(0.5,0.0625,0.0625,0.375)	(0.6,0.1,0.1,0.2)	0.43
1	**(0.53,0.06,0.06,0.35)**	**(0.64,0.1,0.1,0.17)**	0.4

Evolving Trust Distribution (simple)



Evolving Trust Distribution (Robust)



Analysis and Robustness Issues

Salient Properties

- Symmetry
  - Simple Scheme is symmetric w.r.t. trust/experience levels while Robust Scheme is somewhat asymmetric because of non-uniform decay.
  - Experience levels are "preserved" in that extreme/controvertial behavior (*credulous interpretation*) is treated differently from ignorance (*skeptical interpretation*).

Salient Properties

- Effect of Order of Experience
  - Simple Scheme is sensitive to the counts of various experience levels, but not to the order of experience.
  - Robust Scheme is sensitive to the order of experience.

Salient Properties

- Differential Aging of experience levels
  - It exhibits limited and selective memory.
    - –It retains low-level experiences much longer than high-level experiences.
      - »Parameters: Decay rate and saturation

Related Work on Multi-level Trust with Applications

The described approach is similar to Dirichlet Reputation System [Josang-Haller, 2007].

Applications:

- Browser toolbar for clients to see the user ratings and for users to provide ratings (critical surfer model) [Josang-Haller, 2007]
- Evaluating partners in Collaborative Environments [Yang and Cemerlic, 2009]
- Formalizing Multi-Dimensional Contracts [Reece, et al, 2007]
- In Collaborative Intrusion Detection System [Fung et al, 2011]

Conclusion

- Provided simple examples of trust (Why?)
- Explained salient features of trust (What?)
- Showed examples of gleaning trustworthiness (How?)
- Touched upon research challenges in the context of
  - Sensor Networks
  - Social Networks
  - Interpersonal Networks
  - Collaborative Environments

Holy Grail
for Automatic Trust Computation

Develop expressive trust networks that can be assigned objective semantics.

Thank You!


WRIGHT STATE
UNIVERSITY


kno.e.sis


hio Centers of Excellence